

*Una buena ingeniería de software
requiere la diferenciación entre la
especificación y la implementación.*

INGENIERIA DE SOFTWARE

Primer Avance

García Juárez Leonel Alexis

Medina Avendaño Griselda

Ramírez Ramírez Ramiro

Índice general

0.1. Introducción	7
1. Propósito	8
2. Ámbito del Proyecto	9
3. Personal Involucrado	10
4. Factibilidad	11
5. Viabilidad	12
6. Mapa de Navegación	13
7. Requisitos Funcionales	14
8. Interfaces	24
9. Estándar de Codificación	35
9.1. Inicialización	35
9.2. Variables locales	35
9.3. Variables globales	35
9.4. Constantes	35
9.5. Comentarios de inicio o encabezado	36
9.6. Comentarios en línea	36
9.7. Sangrías y tabulaciones	36
9.8. Interfaces	36
10. Diagramas de casos de uso	38
11. Diccionario de datos	56
12. Diagrama Conceptual	59
13. Diseño Lógico	60
14. Diseño Físico	62
15. Diagrama de Paquetes	64
16. Diagrama de secuencias	65

<i>ÍNDICE GENERAL</i>	3
17.Diagramas de Estado	76
18.Diagramas de Actividades	88
19.Diagramas de Flujo de Datos	101
20.Diagrama de Despliegue	104
21.Diagrama de Componentes	106
22.Diagrama de Objetos	108

Índice de figuras

3.1. Roles de usuario	10
6.1. Mapa de navegación	13
7.1. Requisito Funcional 1	14
7.2. Requisito Funcional 2	15
7.3. Requisito Funcional 3	16
7.4. Requisito Funcional 4	17
7.5. Requisito Funcional 5	18
7.6. Requisito Funcional 6	19
7.7. Requisito Funcional 7	20
7.8. Requisito Funcional 8	21
7.9. Requisito Funcional 9	22
7.10. Requisito Funcional 10	23
8.1. Login	24
8.2. Pantalla Principal	25
8.3. Productos	26
8.4. Nuevo Producto	27
8.5. Modificar Producto	28
8.6. Eliminar Producto	29
8.7. Inventario	30
8.8. Ventas	31
8.9. Reporte	32
8.10. Corte	33
8.11. Usuarios	34
10.1. CU Login	39
10.2. Diagrama de CU Ventas	40
10.3. CU Ventas	41
10.4. Diagrama de CU Nuevo Producto	42
10.5. CU Nuevo Producto	43
10.6. Diagrama de CU Modificar Producto	44
10.7. CU Modificar Producto	45
10.8. Diagrama de CU Eliminar Producto	46
10.9. CU Eliminar Producto	47
10.10. Diagrama de CU Inventario	48
10.11. CU Inventario	49

10.12	Diagrama de CU Corte	50
10.13	CU Corte	51
10.14	Diagrama de CU Reporte	52
10.15	CU Reporte	53
10.16	Diagrama de CU Usuarios	54
10.17	CU Uusario	55
11.1.	Productos	56
11.2.	Ventas	57
11.3.	Categorías	57
13.1.	Diagrama ERD	60
13.2.	Esquema Relacional	61
14.1.	Script	62
14.2.	Script1.1	63
15.1.	Diagrama de Paquetes	64
16.1.	Login	68
16.2.	Login1.1	69
16.3.	Nuevo Producto	70
16.4.	Consultar	71
16.5.	Eliminar	72
16.6.	Modificar Producto	73
16.7.	Inventario	73
16.8.	Corte	74
16.9.	Reporte	75
16.10	Ventas	75
17.1.	Login	79
17.2.	Nuevo	80
17.3.	Modificar	81
17.4.	Eliminar	82
17.5.	Consultar	83
17.6.	Inventario	84
17.7.	Corte	85
17.8.	Reporte	86
17.9.	Ventas	87
18.1.	Login	92
18.2.	Nuevo	93
18.3.	Modificar	94
18.4.	Eliminar	95
18.5.	Consultar	96
18.6.	Inventario	97
18.7.	Corte	98
18.8.	Reporte	99
18.9.	Ventas	100
19.1.	Nivel 0	101

19.2. Nivel 0.1 102

19.3. Nivel 1 103

20.1. Ferretería 105

21.1. Ferretería 107

22.1. Ferretería 110

.

0.1. Introducción

Este proyecto ha sido elaborado dada la necesidad de automatizar un proceso de ventas de una ferretería, la cual requiere generar informes exactos sobre las ventas diarias, además de que requiere de llevar un control de inventario de productos para poder supervisar el stock disponible de los mismos, además se pretende que a través de este sistema los empleados mejoren sus tiempos de atención al cliente en ventas ya que evitarían hacer cuentas de forma manual.

El sistema de software que implementaremos facilitará el manejo de ventas, inventario, ingreso de productos, etc ya que esto se realizará de manera sistematizada ya que actualmente se realiza de manera manual, por lo anterior de este sistema ayudaría a visualizar el rendimiento de la empresa tomando en cuenta las ventas generadas por el sistema.

Este proyecto será realizado con el lenguaje de programación Java, además de que teniendo en cuenta que es un sistema que va a generar una gran cantidad de datos debido a las diferentes funcionalidades del mismo, se pretende hacer uso de una base de datos que los almacene. Como se mencionó antes este sistema contará con distintas funcionalidades las cuales ayudarían a agilizar tareas como la generación de reportes de ventas por días, generación de corte diario, manejo de ventas por cliente, cobro por ventas, y el manejo del inventario.

La realización del estudio de métodos en la ferretería es importante debido a que partir de este, se puede realizar una mejora a los procesos estudiados, haciendo que sean más óptimos y que a su vez permitan aumentar los niveles de productividad de la compañía.

Capítulo 1

Propósito

El objetivo de este proyecto constituye una herramienta que permite un manejo oportuno y eficiente de la información, siendo posible el rendimiento de tiempo respecto a sus productos. Al igual que permita lograr la eficiencia en el control de los productos con los que se va a comercializar, como también llevar un control del inventario, registro de ventas en productos de una manera más rápida, organizada y sencilla beneficiando tanto al personal como a los clientes de la empresa. Emitirá reportes sobre ventas de manera eficaz, de forma que se logre un buen manejo de las transacciones que incurren en la empresa.

Capítulo 2

Ámbito del Proyecto

La ferretería a la cual está destinado el proyecto, desea un sistema de administración para ingreso y salida de mercancía sobre control del inventario y ventas, que ayude a mejorar la administración y control de la empresa, ya que al contar con un sistema manual se tienen dificultades para obtener de forma ágil y rápida la información relevante sobre las existencias. El proyecto consiste en diseñar y construir un sistema de interfaz amigable que permita llevar un control sobre los productos disponibles, además de mantener un registro de las salidas de los productos, de tal forma que esta información sirva al propietario del negocio para tomar decisiones como cuando reabastecerse de productos, también se pretende que el sistema gestione las ventas realizando su registro y generando un reporte de las mismas cuando este se requiera. Para el desarrollo de este proyecto se hará uso de distintas técnicas de programación aprendidas durante el curso como programación orientada a objetos, este paradigma nos servirá para la creación de la interfaz gráfica de usuario así como la funcionalidad deseada, combinando el lenguaje de programación Java con el lenguaje de consulta de mysql para poder manipular de mejor manera los datos.

Capítulo 3

Personal Involucrado

Nombre	Rol de usuario
Everardo de Jesus Pacheco Antonio	Administrador
Rolando Pedro Gabriel	Administrador
Lirio Ruiz Guerra	Empleado
Ramiro Ramírez Ramírez	Administrador
Leonel Alexis García Juárez	Administrador
Griselda Medina Avendaño	Administrador

Figura 3.1: Roles de usuario

Capítulo 4

Factibilidad

Es posible realizar este proyecto dado que para la factibilidad técnica, el equipo de desarrollo cuenta con conocimientos en programación en java así como el manejo de base de datos, además de que se cuenta con herramientas como el entorno de desarrollo Netbeans de cual se hará uso, además del sistema gestor de base de datos Postgres. En cuanto a la factibilidad operativa, se cuenta con usuarios finales dispuestos a emplear las funcionalidades generadas por el sistema que se desarrollará.

Capítulo 5

Viabilidad

Este proyecto es viable ya que contamos con conocimiento acerca de las distintas herramientas de desarrollo de software (Java, Postgres,MySQL). Además de esto cabe recalcar que contamos con las herramientas necesarias de trabajo, como Ide's para formar la base de nuestro sistema que esta destinado a una ferretería. Se cuenta con el equipo necesario para lograr el objetivo. También nos encontramos organizados como equipo, y esto hace que el trabajo sea más sencillo de realizar, ya que cada uno aporta distintas ideas en cuanto a diseño, lógica y redacción de los distintos aspectos que requiera el sistema. El sistema será viable ya que se cuenta con los recursos necesarios que se requiera durante el proceso del mismo.

En cuanto al aspecto económico del sistema, no tenemos contemplado recurrir a gastos innecesarios para material externo, porque contamos con el material y herramientas adecuadas para la realización. La ventaja que se tiene es que cada integrante cuenta con un recurso material (laptos) para realizar las bases del sistema, por la parte de licencias de software o algún pago externo se ocupara software libre, gracia a que la herramienta Java es multiplataforma se evitara costes para recurrir a esto.

Capítulo 6

Mapa de Navegación

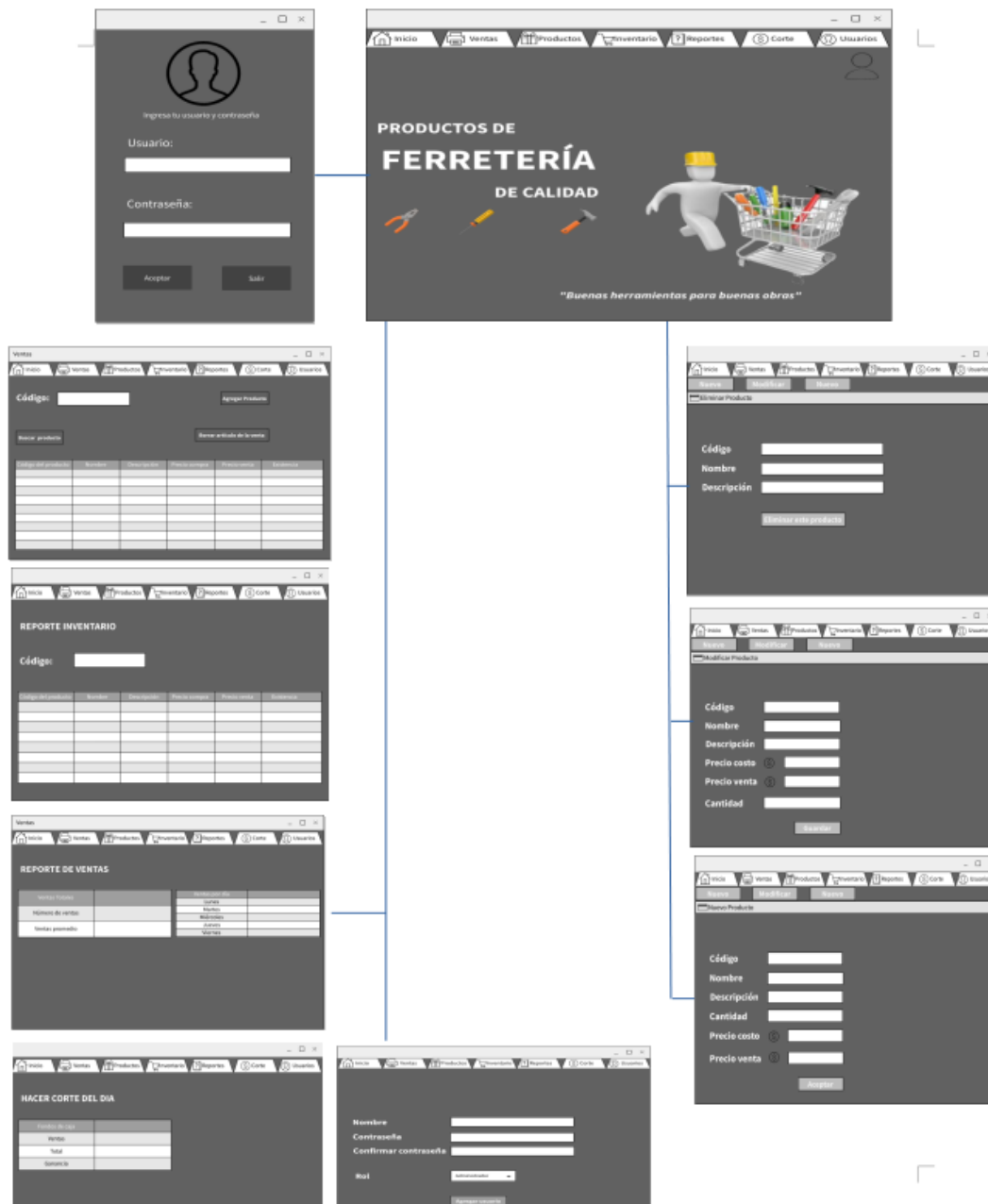


Figura 6.1: Mapa de navegación

Capítulo 7

Requisitos Funcionales

IDENTIFICADOR RF1		NOMBRE Buscar Producto	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Producto		SALIDA Producto con todos sus datos (descripción, precio, existencia)	
DESCRIPCIÓN PRECONDICIÓN: Se desea localizar un producto por nombre o código. DESCRIPCIÓN: Se verifica el nombre del producto y el código. POSTCONDICIÓN: Se localiza el producto y se muestran sus datos.			
MANEJO DE SITUACIONES ANORMALES 1. Producto inexistente			
CRITERIOS DE ACEPTACIÓN 1. EL código ingresado es correcto, se encuentra el producto.			

Figura 7.1: Requisito Funcional 1

IDENTIFICADOR RF2		NOMBRE Agregar Producto a la venta	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código		SALIDA Se agrega producto a la venta	
DESCRIPCIÓN PRECONDICIÓN: Se requiere localizar producto para agregar a la venta. DESCRIPCIÓN: Se busca producto deseado POSTCONDICIÓN: Se agrega producto localizado a la venta.			
MANEJO DE SITUACIONES ANORMALES 1.Producto inexistente, se envía mensaje de error al usuario.			
CRITERIOS DE ACEPTACIÓN 1. EL código ingresado es correcto, se localiza el producto.			

Figura 7.2: Requisito Funcional 2

IDENTIFICADOR RF3		NOMBRE Borrar articulo de la venta	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Selección del producto		SALIDA Producto eliminado de la venta	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con un producto de la venta seleccionado. DESCRIPCIÓN: Se consulta al usuario si quiere eliminar el artículo. POSTCONDICIÓN: Se elimina el producto de la venta.			
MANEJO DE SITUACIONES ANORMALES			
CRITERIOS DE ACEPTACIÓN 1. Se selecciona producto de la venta y se elimina de la misma.			

Figura 7.3: Requisito Funcional 3

IDENTIFICADOR RF4		NOMBRE Nuevo Producto	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Descripción Cantidad Precio costo Precio venta		SALIDA Alta del producto en el sistema	
DESCRIPCIÓN PRECONDICIÓN: No se cuenta con registro del producto en sistema. DESCRIPCIÓN: Se ingresan los datos del producto. POSTCONDICIÓN: Producto es dado de alta en el sistema.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error y no se reconoce el nuevo producto).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se agrega un nuevo producto.			

Figura 7.4: Requisito Funcional 4

IDENTIFICADOR RF5		NOMBRE Modificar Producto	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Descripción Precio Costo Precio venta Cantidad		SALIDA Datos de producto son modificados	
DESCRIPCIÓN PRECONDICIÓN: Se desea modificar algún dato del producto. DESCRIPCIÓN: Se ingresan los datos y el código del producto. POSTCONDICIÓN: Se modifica el producto.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se modifica el producto.			

Figura 7.5: Requisito Funcional 5

IDENTIFICADOR RF6		NOMBRE Eliminar Producto	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código		SALIDA Producto eliminado del sistema	
DESCRIPCIÓN PRECONDICIÓN: Se desea dar de baja un producto, eliminar todos sus datos. DESCRIPCIÓN: Se ingresa código o nombre del producto para su localización. POSTCONDICIÓN: Se elimina producto del inventario.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se elimina el producto.			

Figura 7.6: Requisito Funcional 6

IDENTIFICADOR RF7		NOMBRE Reporte de inventario	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Inventario		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Productos almacenados en la base de datos		SALIDA Reporte de inventario.	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con productos dados de alta. DESCRIPCIÓN: Se ingresa a la sección inventario. POSTCONDICIÓN: Se genera un reporte de inventario donde se pueden ver todos los productos disponibles con sus datos correspondientes.			
MANEJO DE SITUACIONES ANORMALES 1. No se han dado de alta productos en sistema (Se mostrará en pantalla un mensaje de inexistentes).			
CRITERIOS DE ACEPTACIÓN 1. Existen productos dados de alta, se genera el reporte de inventario.			

Figura 7.7: Requisito Funcional 7

IDENTIFICADOR RF8		NOMBRE Reporte de ventas	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección reportes		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Ventas totales realizadas en una semana		SALIDA Reporte de: Ventas totales Número de ventas Venta promedio Ventas por día	
DESCRIPCIÓN PRECONDICIÓN: Se han realizado ventas DESCRIPCIÓN: Se consulta el importe total generado por cada venta. POSTCONDICIÓN: Se genera el reporte de ventas.			
MANEJO DE SITUACIONES ANORMALES 1. No se han realizado ninguna venta en la semana(Se mostrará en pantalla un mensaje).			
CRITERIOS DE ACEPTACIÓN 1. Se han realizado ventas, se genera el reporte de ventas.			

Figura 7.8: Requisito Funcional 8

IDENTIFICADOR RF9		NOMBRE Hacer corte del día	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección corte		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Importe total de las ventas realizadas en el día Fondo de caja inicial		SALIDA Visualización de: Fondo de caja Ventas Total Ganancia	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con ventas realizadas en el día. DESCRIPCIÓN: Se realizan cálculos para mostrar total de ventas y ganancias. POSTCONDICIÓN: Se genera el corte del día.			
MANEJO DE SITUACIONES ANORMALES 1. No se ha realizado ni una venta en el día (Se mostrará en pantalla un mensaje al usuario).			
CRITERIOS DE ACEPTACIÓN 1. Se han realizado ventas en el día, se genera corte del día.			

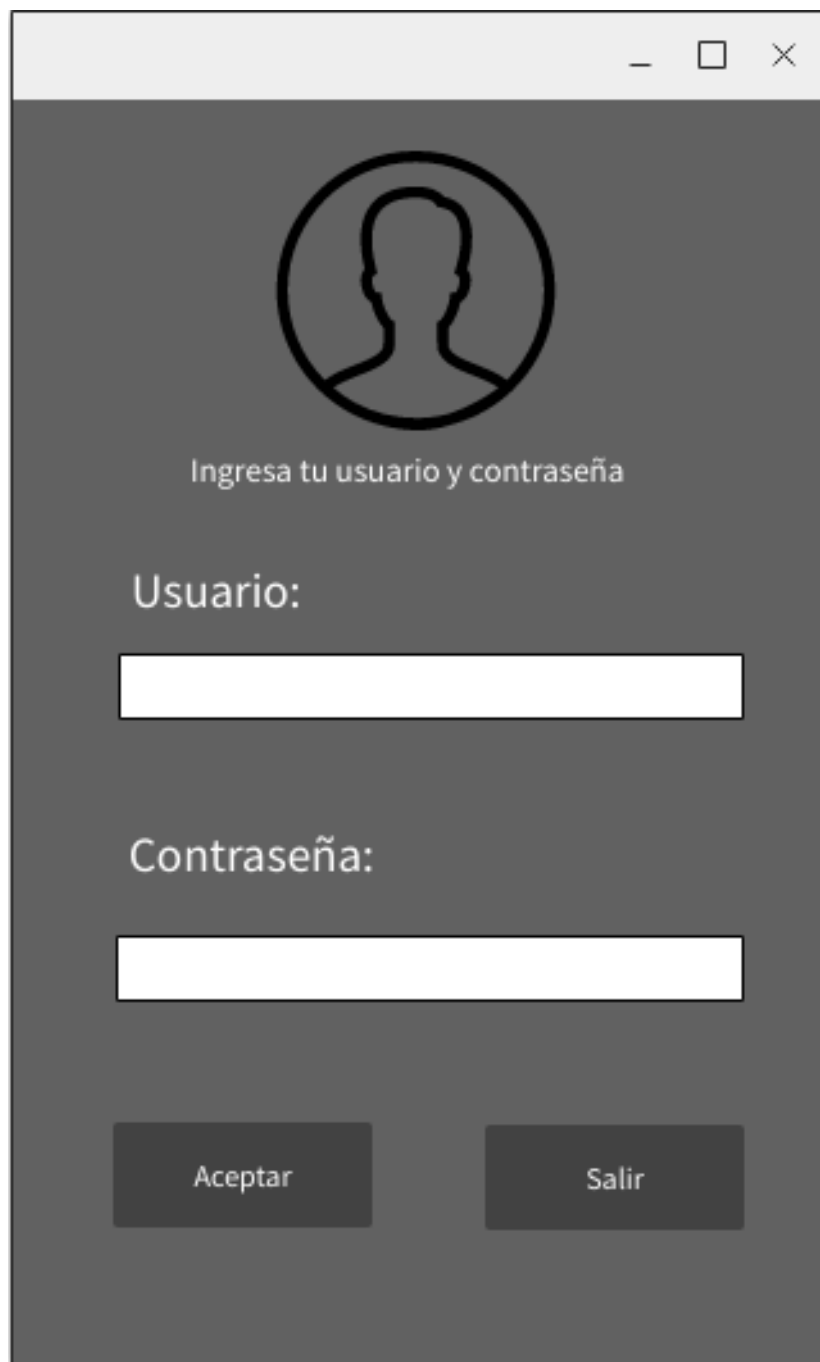
Figura 7.9: Requisito Funcional 9

IDENTIFICADOR RF10		NOMBRE Agregar Usuario	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección Usuarios		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre Contraseña Rol		SALIDA Registro de nuevo usuario.	
DESCRIPCIÓN PRECONDICIÓN: No se cuenta con registro de usuario en el sistema. DESCRIPCIÓN: Se ingresan los datos del usuario a registrar. POSTCONDICIÓN: Se registra el nuevo usuario.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error y no se ingresa el usuario).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se registra un nuevo usuario.			

Figura 7.10: Requisito Funcional 10

Capítulo 8

Interfaces



A login window with a dark gray background and a light gray title bar. The title bar contains standard window controls (minimize, maximize, close). The main content area features a circular icon of a person's silhouette. Below the icon is the text "Ingresa tu usuario y contraseña". There are two input fields: "Usuario:" and "Contraseña:". At the bottom, there are two buttons: "Aceptar" and "Salir".

Ingresa tu usuario y contraseña

Usuario:

Contraseña:

Aceptar Salir

Figura 8.1: Login



Figura 8.2: Pantalla Principal

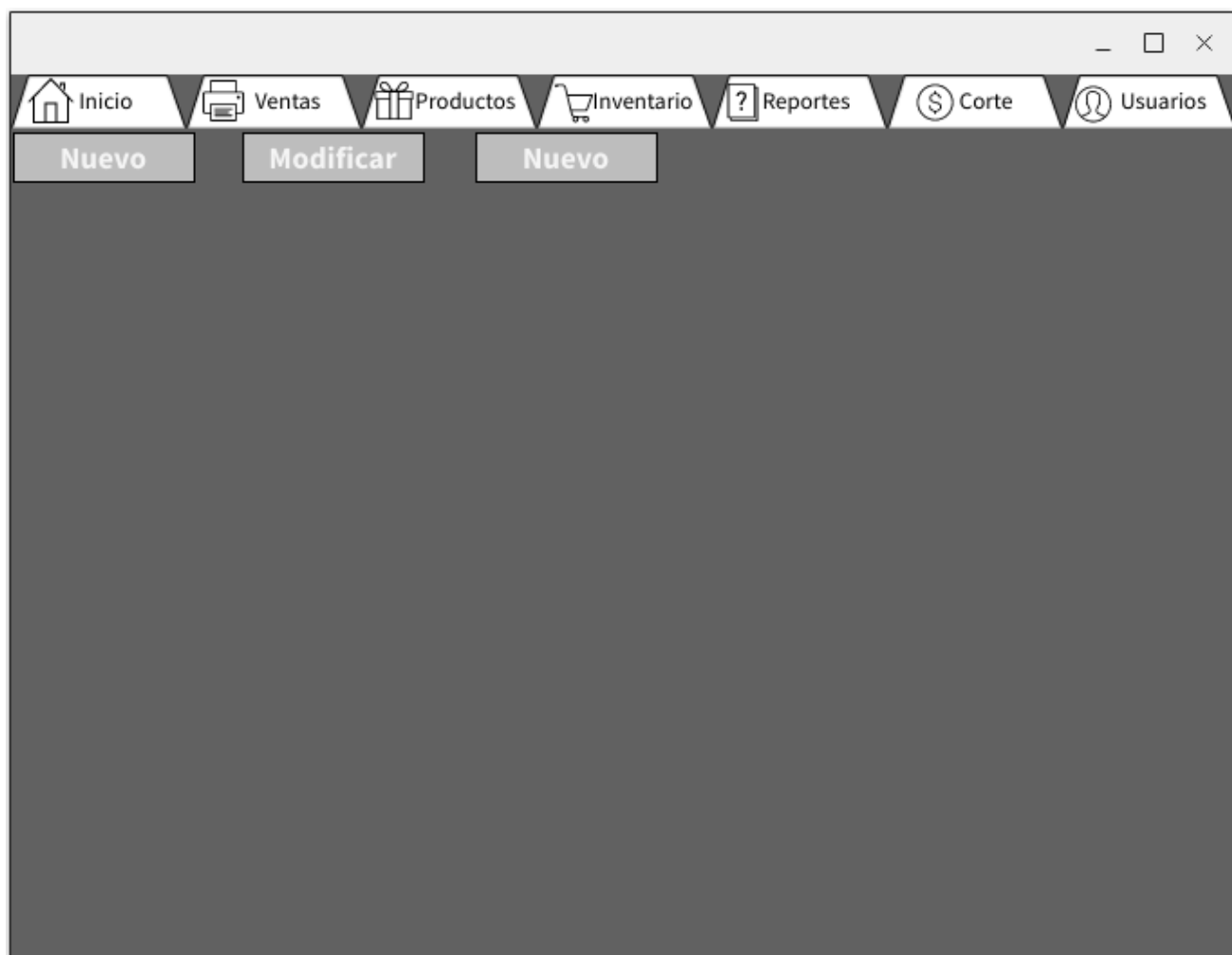


Figura 8.3: Productos

The image shows a software window titled "Nuevo Producto". At the top, there is a navigation bar with icons and labels: "Inicio" (house icon), "Ventas" (printer icon), "Productos" (gift icon), "Inventario" (shopping cart icon), "Reportes" (question mark icon), "Corte" (dollar sign icon), and "Usuarios" (person icon). Below this bar are three buttons: "Nuevo", "Modificar", and "Nuevo". The main area of the window contains a form with the following fields and labels:

- Código**: A text input field.
- Nombre**: A text input field.
- Descripción**: A text input field.
- Cantidad**: A text input field.
- Precio costo**: A label followed by a dollar sign icon and a text input field.
- Precio venta**: A label followed by a dollar sign icon and a text input field.

At the bottom center of the form is a button labeled "Aceptar".

Figura 8.4: Nuevo Producto

The image shows a web application interface for modifying a product. At the top, there is a navigation bar with icons and labels for 'Inicio', 'Ventas', 'Productos', 'Inventario', 'Reportes', 'Corte', and 'Usuarios'. Below this, there is a secondary bar with buttons labeled 'Nuevo', 'Modificar', and 'Nuevo'. The main content area is titled 'Modificar Producto' and contains several input fields for product details: 'Código', 'Nombre', 'Descripción', 'Precio costo' (with a dollar sign icon), 'Precio venta' (with a dollar sign icon), and 'Cantidad'. A 'Guardar' button is located at the bottom right of the form.

Código	<input type="text"/>
Nombre	<input type="text"/>
Descripción	<input type="text"/>
Precio costo	<input type="text"/>
Precio venta	<input type="text"/>
Cantidad	<input type="text"/>

Figura 8.5: Modificar Producto

The image shows a web application window with a navigation bar at the top. The navigation bar contains several tabs: 'Inicio' (Home), 'Ventas' (Sales), 'Productos' (Products), 'Inventario' (Inventory), 'Reportes' (Reports), 'Corte' (Cut), and 'Usuarios' (Users). Below the navigation bar, there are three buttons: 'Nuevo' (New), 'Modificar' (Modify), and 'Nuevo' (New). A sub-header bar contains a trash icon and the text 'Eliminar Producto' (Delete Product). The main content area is dark gray and contains three input fields labeled 'Código' (Code), 'Nombre' (Name), and 'Descripción' (Description). Below these fields is a button labeled 'Eliminar este producto' (Delete this product).

Inicio Ventas Productos Inventario Reportes Corte Usuarios

Nuevo Modificar Nuevo

Eliminar Producto

Código

Nombre

Descripción

Eliminar este producto

Figura 8.6: Eliminar Producto

The screenshot shows a web application window with a navigation bar at the top containing icons and labels for 'Inicio', 'Ventas', 'Productos', 'Inventario', 'Reportes', 'Corte', and 'Usuarios'. The 'Reportes' section is active. Below the navigation bar, the title 'REPORTE INVENTARIO' is displayed. A label 'Código:' is followed by an empty text input field. Below this is a table with six columns: 'Código del producto', 'Nombre', 'Descripción', 'Precio compra', 'Precio venta', and 'Existencia'. The table contains ten empty rows for data entry.

Código del producto	Nombre	Descripción	Precio compra	Precio venta	Existencia

Figura 8.7: Inventario

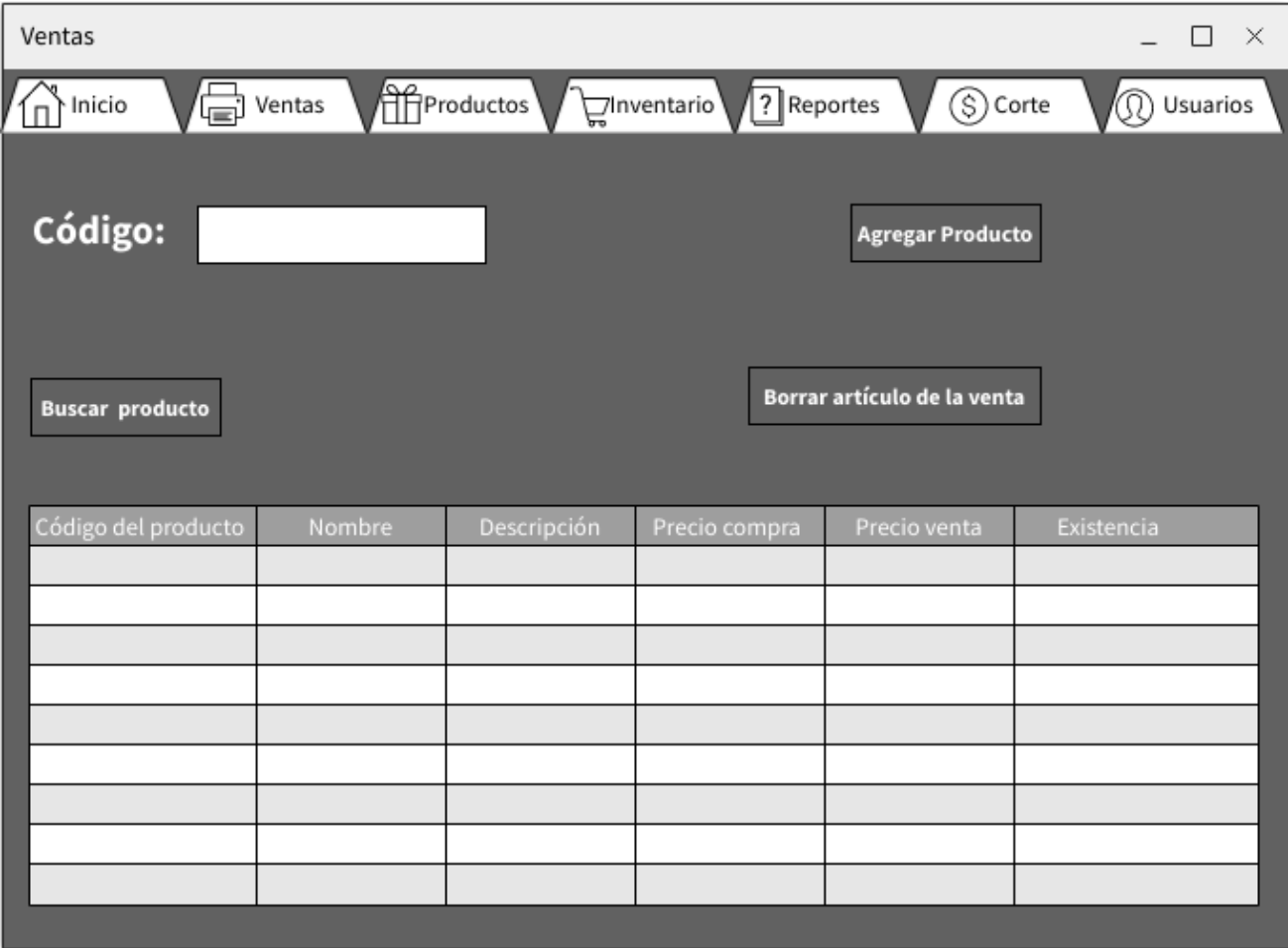


Figura 8.8: Ventas



Figura 8.9: Reporte

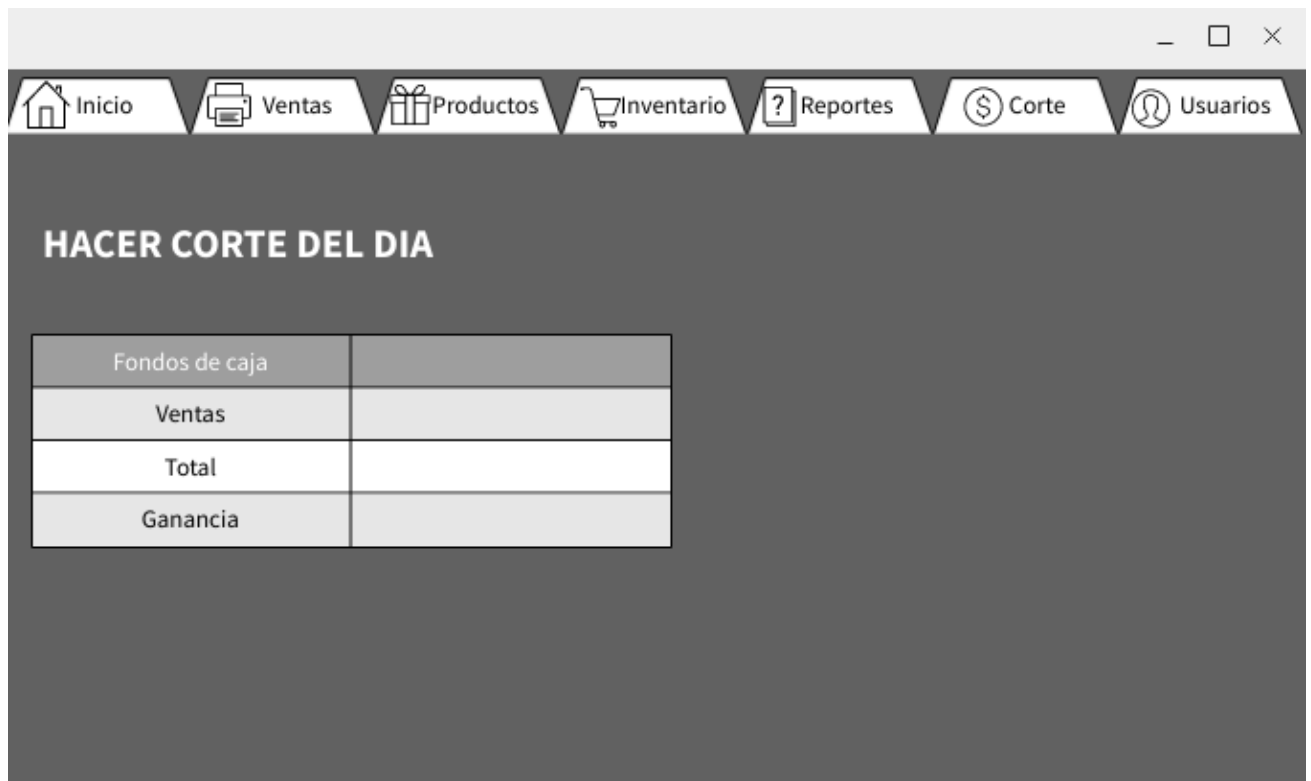


Figura 8.10: Corte

The image shows a web application window with a dark gray background. At the top, there is a navigation bar with seven items: 'Inicio' (with a house icon), 'Ventas' (with a printer icon), 'Productos' (with a gift icon), 'Inventario' (with a shopping cart icon), 'Reportes' (with a question mark icon), 'Corte' (with a dollar sign icon), and 'Usuarios' (with a person icon). Below the navigation bar, the main content area contains a form for adding a user. The form has four labels on the left: 'Nombre', 'Contraseña', 'Confirmar contraseña', and 'Rol'. Each label is followed by a corresponding input field. The 'Nombre' field is a simple text box. The 'Contraseña' and 'Confirmar contraseña' fields are also simple text boxes. The 'Rol' field is a dropdown menu with 'Administrador' selected. Below the 'Rol' dropdown is a button labeled 'Agregar usuario'.

Nombre

Contraseña

Confirmar contraseña

Rol

Figura 8.11: Usuarios

Capítulo 9

Estándar de Codificación

[T1]fontenc[utf8]inputenc

9.1. Inicialización

Toda variable local deberá ser inicializada en el momento de declaración, deberá de estar situada al principio de cada bloque principal.

9.2. Variables locales

Deben estar situadas en el principio de cada bloque principal:

```
public class Ejemplo
{
    public static void main(String[] args)
    {
        //Declaración de variables
        int contador=0;
        double r=0.0;
    }
}
```

9.3. Variables globales

Estas deben estar ubicadas al inicio de la clase principal.

Notación

Se usará la notación CamelCase para la declaración de métodos o variables. //ejemplo de un método definido con CamelCase en CalcularSuma public int CalcularSuma(int b, int a) int resultado; resultado=b+a; return resultado;

9.4. Constantes

La declaración de constantes deberá ser declarado con letras mayúsculas.

```
public class Ejemplo
{
    public static void main(String[] args)
    {
        //Declaración de constantes
        static final int DIASSEMANA = 7;
    }
}
```

```
static final int DIASLABORABLES = 5;
```

9.5. Comentarios de inicio o encabezado

Se usará un encabezado al inicio de cada bloque o clase creada, también deberá ser asignada a las interfaces. El encabezado usado hará uso de comentarios en bloque como se muestra a continuación:

```
/** @Autor: */
/* Descripción del proyecto: */
/* Fecha de creación: */
/* Fecha de modificación: */
package ejemplo;
public class Ejemplo
public static void main(String[] args)
//Código
```

9.6. Comentarios en línea

Estos comentarios se deberán hacer en una línea, con cierto espacio si se quiere documentar la función de alguna operación o método.

```
public int CalcularSuma(int b, int a)
int resultado; //Se declara una variable de tipo entero
resultado=b+a; //Calcula la suma de b+a y la guarda en la variable resultado
return resultado; //Retorna la variable resultado
```

9.7. Sangrías y tabulaciones

Las tabulaciones serán igual a cuatro espacios, mientras que las sangrías deberán de mostrar claramente los procesos mas internos del código.

```
public class Ejemplo
    public static void main(String[] args)
        int num=0;
        if(num==0)
            System.out.println("Número igual a cero");

        else if(num>0)
            System.out.println("Número mayor que cero");

        else
            System.out.println("Número menor que cero");
```

9.8. Interfaces

Para los nombres que serán usados en los componentes de las interfaces se agregará un identificador, esto con el fin de reconocer mas fácil a los componentes. Los componentes que se usaran serán los

siguientes:

```
jTextField txtEjemplo  
jLabel lblEjemplo  
jButton btnEjemplo  
jList lsEjemplo  
jRadioButton rbtnEjemplo  
jCheckBox cbEjemplo  
jComboBox cbxEjemplo  
jPanel jpEjemplo  
jTabbedPane jtEjemplo  
jPasswordField jpsEjemplo  
jTable tbEjemplo
```

Capítulo 10

Diagramas de casos de uso

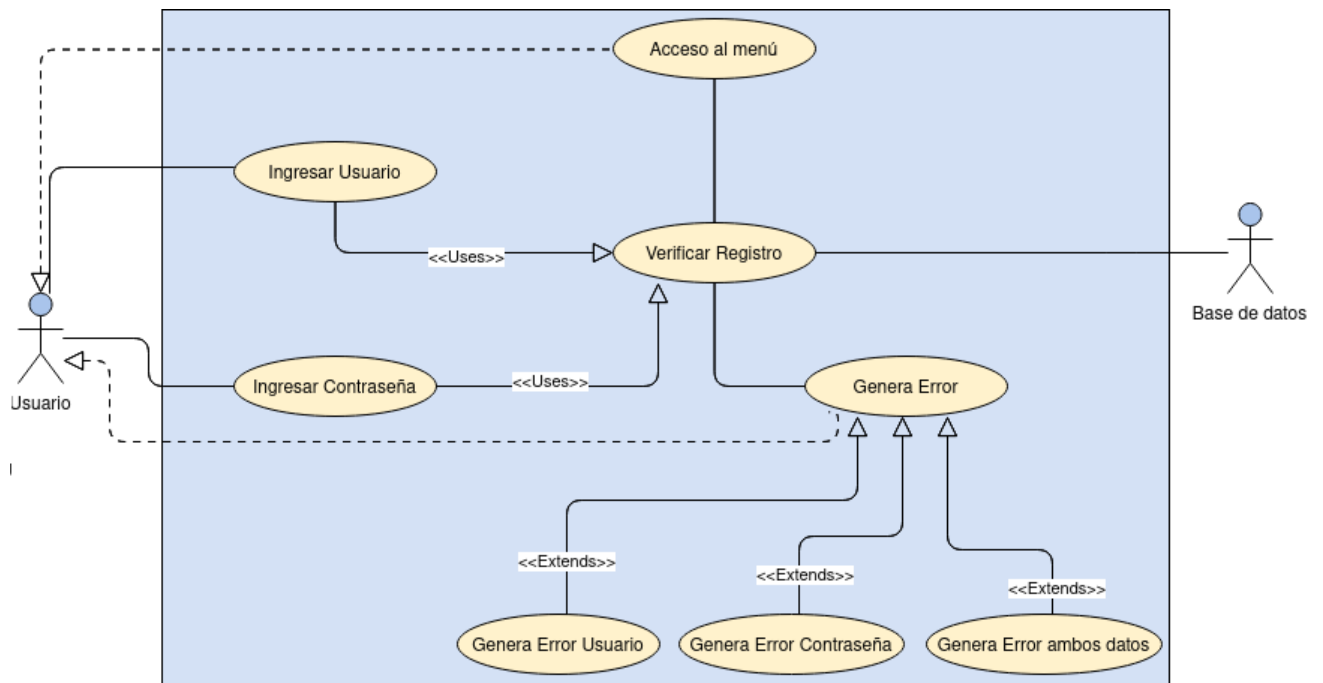


Diagrama de CU Login

ID:	CU-1		
Nombre:	Login		
Creado por:	GJLA	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	01/12/2021
Actores:	Administrador, Usuario, Base de datos		
Descripción:	Conceder acceso del usuario al sistema.		
Disparador:	N/A		
Precondiciones:	N/A		
Postcondiciones:	Se accede a la pantalla inicio.		
Flujo normal:	<p>1. Ingresar nombre.</p> <p>2. Ingresar contraseña.</p> <p>3. Se presiona el botón aceptar.</p> <p>3.1 En caso de que nombre de usuario no exista se envía mensaje de error “Usuario erróneo”.</p> <p>3.2 En caso de que contraseña sea incorrecta se envía mensaje de error: “Contraseña incorrecta”.</p> <p>3.3 Si no se ha ingresado uno de los dos datos (nombre o contraseña) se envía mensaje de error: “Uno de los campos esta vacío”.</p> <p>2. El sistema valida los datos ingresados en 1 y 2.</p> <p>2.1 En caso de que los datos sean correctos se concede el acceso, y se remite a la pantalla “Inicio”.</p> <p>2.1 En caso de datos inválidos (nombre o contraseña) se envía mensaje de error al usuario.</p>		
Flujo alterno:	Una vez que se concede el acceso, usuario accede a la pantalla “inicio” con el menú de opciones que le corresponda según su rol(administrador o usuario).		
Incluye:	N/A		
Frecuencia de uso:	Alta		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.1: CU Login

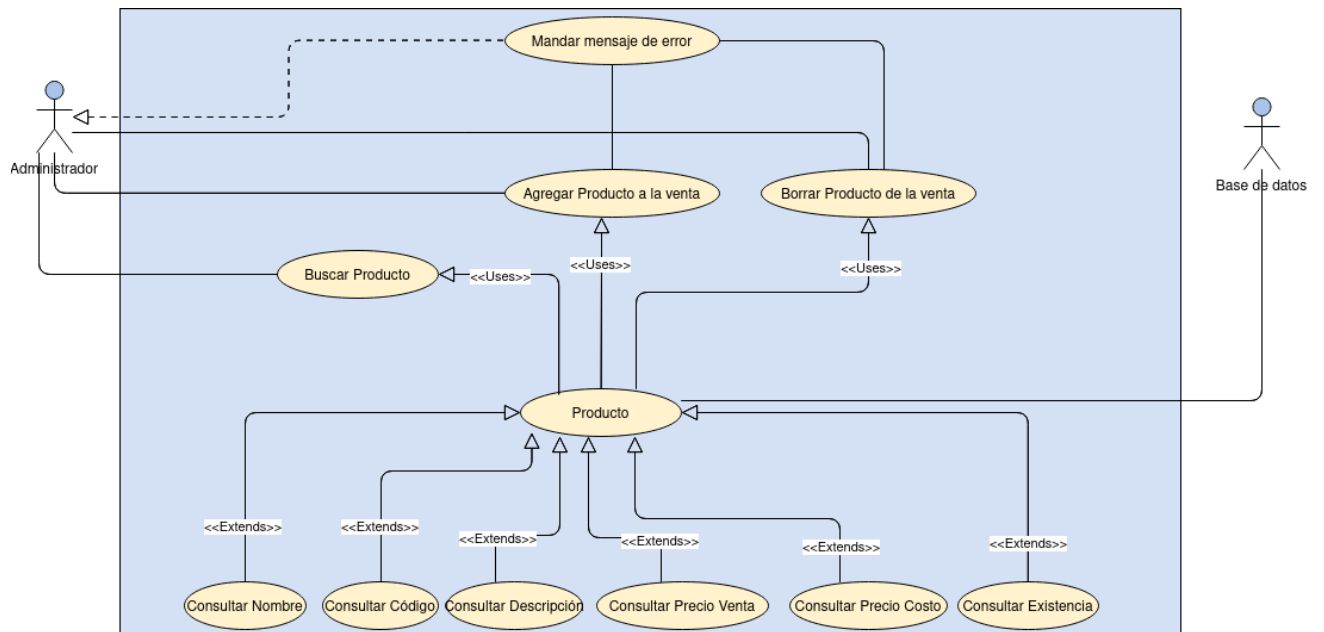


Figura 10.2: Diagrama de CU Ventas

ID	CU-2		
Nombre	Ventas de producto		
Creado por	RRR	Actualizado por:	RRR
Fecha de creación	30/10/2021	Fecha de última revisión:	30/10/2021
Actores	Administrador, usuario, base de datos		
Descripción	Registrar ventas y cobrar (Pantalla).		
Disparador	Se selecciona la opción “Ventas” del menú principal		
Precondiciones	Usuario debe haberse logeado en el sistema.		
Postcondiciones	Se reinician los valores y se muestra la pantalla ().		
Flujo normal	<ol style="list-style-type: none"> 1. Ingresar el código de barras del producto. <ol style="list-style-type: none"> 1.1 Si el código no existe, se envía un mensaje de “Error” y vuelve al punto 1, para ingresar un nuevo código. 2. El sistema muestra las características del producto en una tabla. <ol style="list-style-type: none"> 2.1 Si el usuario desea registrar otra venta ir al punto 1. <ol style="list-style-type: none"> 2.1.1 Se continua con el flujo. 3. Presionar el botón cobrar. 		
Flujo alternativo	Al terminar de agregar los productos ir a la pantalla ().		
Includes	N/A		
Frecuencia de uso	Muy frecuente		
Requerimientos especiales	N/A		
Supuestos	N/A		
Isues y Notas	N/A		

Figura 10.3: CU Ventas

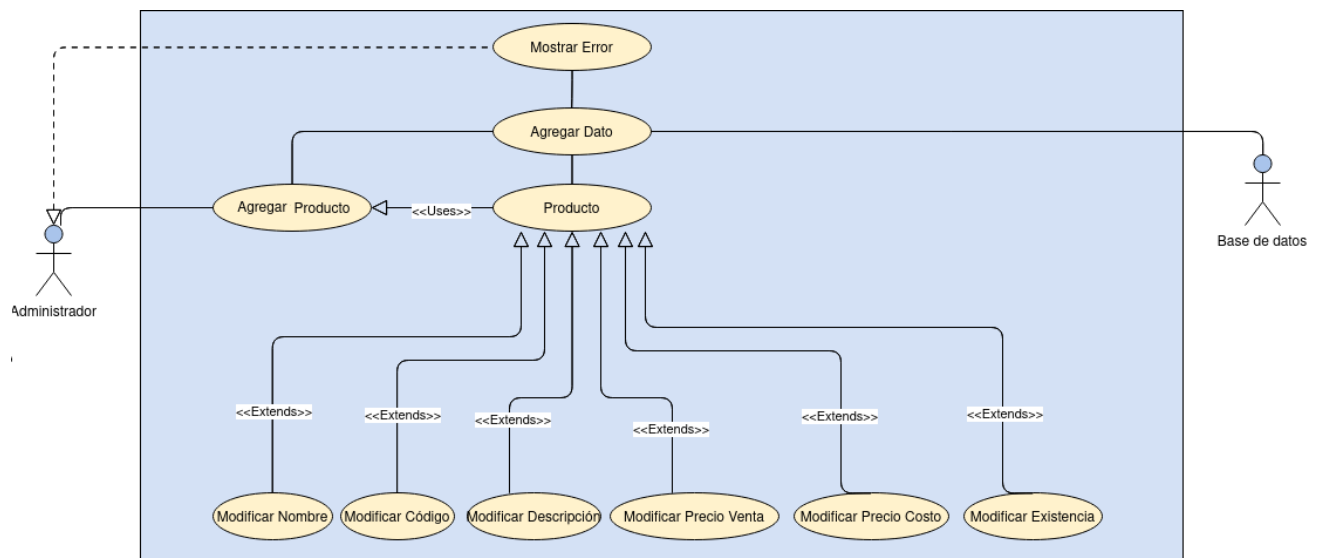


Figura 10.4: Diagrama de CU Nuevo Producto

ID:	CU-3		
Nombre:	Nuevo Producto		
Creado por:	RRR	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	03/12/2021
Actores:	Administrador, usuario, base de datos		
Descripción:	Registra nuevos productos con sus datos correspondientes en el sistema.		
Disparador:	Se selecciona la opción “Productos” del menú y enseguida presionar el botón “Nuevo” .		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se refresca la pantalla para que el usuario agregue nuevo producto si lo desea.		
Flujo normal	<p>1. Se ingresan datos del producto: Nombre, descripción, precio de compra, precio de venta, cantidad.</p> <p>1.1 El sistema valida los datos; el nombre y descripción sean ingresados con caracteres alfabéticos, en caso de ser diferentes no permite el ingreso de datos.</p> <p>1.2 El sistema valida que el precio de compra y precio de venta sean valores float, en caso de ser diferentes no se permite el ingreso de los datos.</p> <p>1.3 El sistema valida que la cantidad sea un valor entero en caso de ser diferente no se permite en ingreso de datos en pantalla.</p> <p>2. Pulsa el botón aceptar y se guarda el producto en la base de datos.</p> <p>2.1 En caso de que un campo este vacío se envía mensaje de error: “Uno de los campos esta vacío”.</p>		
Flujo alterno:	Al terminar de agregar los productos se refresca pantalla.		
Includes:	N/A		
Frecuencia de uso:	Muy frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.5: CU Nuevo Producto

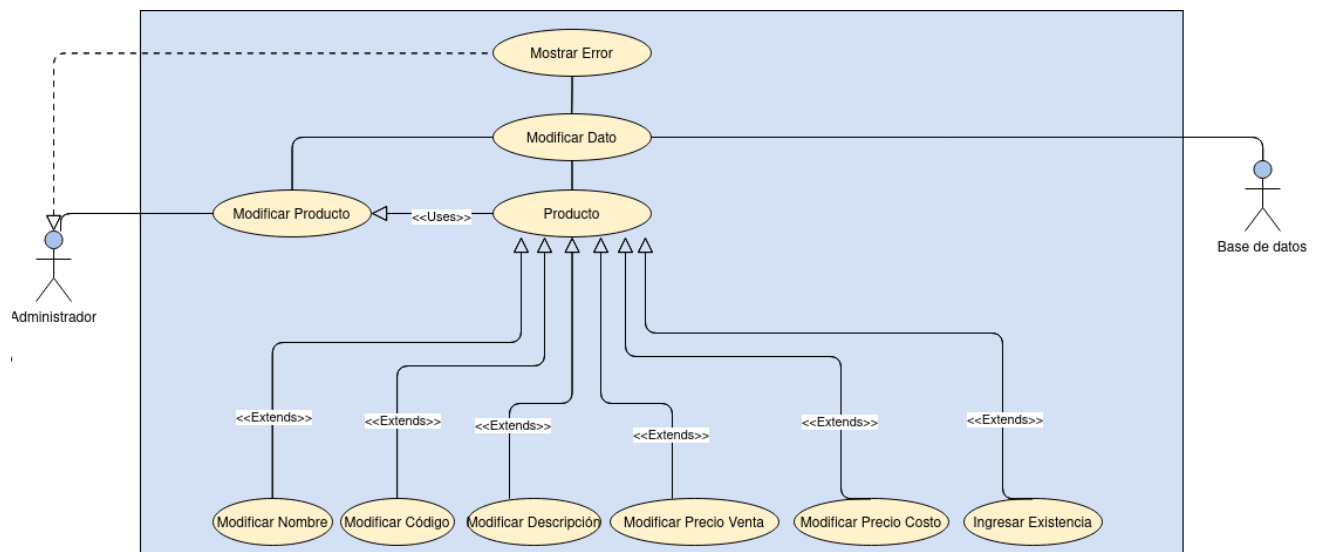


Figura 10.6: Diagrama de CU Modificar Producto

ID:	CU-4		
Nombre:	Modificar producto		
Creado por:	MAG	Actualizado por:	MAG
Fecha de creación:	30/10/2021	Fecha de última revisión:	01/12/2021
Actores:	Administrador, usuario, base de datos		
Descripción:	Realiza cambios a los datos del producto.		
Disparador:	Se selecciona la opción “Productos” del menú y enseguida presionar el botón “Modificar” .		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se actualiza la información del producto en la base de datos.		
Flujo normal:	<ol style="list-style-type: none"> 1. Se ingresa código del producto. 2. Se muestran en pantalla los datos del producto que coincidan con el código ingresado. 3. Usuario edita campo del producto que desea actualizar. <ol style="list-style-type: none"> 3.1 Si los datos del nombre y descripción no corresponden a caracteres alfabéticos, no se permite el ingreso de los datos. 3.2 Si los datos precio compra y precio venta no corresponden a valores float no se permite el ingreso de los datos. 3.3 Si el dato cantidad no corresponde a un valor entero no se permite el ingreso de dicho dato. 4. Una vez que el usuario ha ingresado todos los datos que desea cambiar selecciona la opción “guardar”. 5. Se actualizan datos en la base de datos. 		
Flujo alterno:	Al terminar de agregar los productos se refresca la pantalla.		
Includes:	N/A		
Frecuencia de uso:	Muy frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.7: CU Modificar Producto

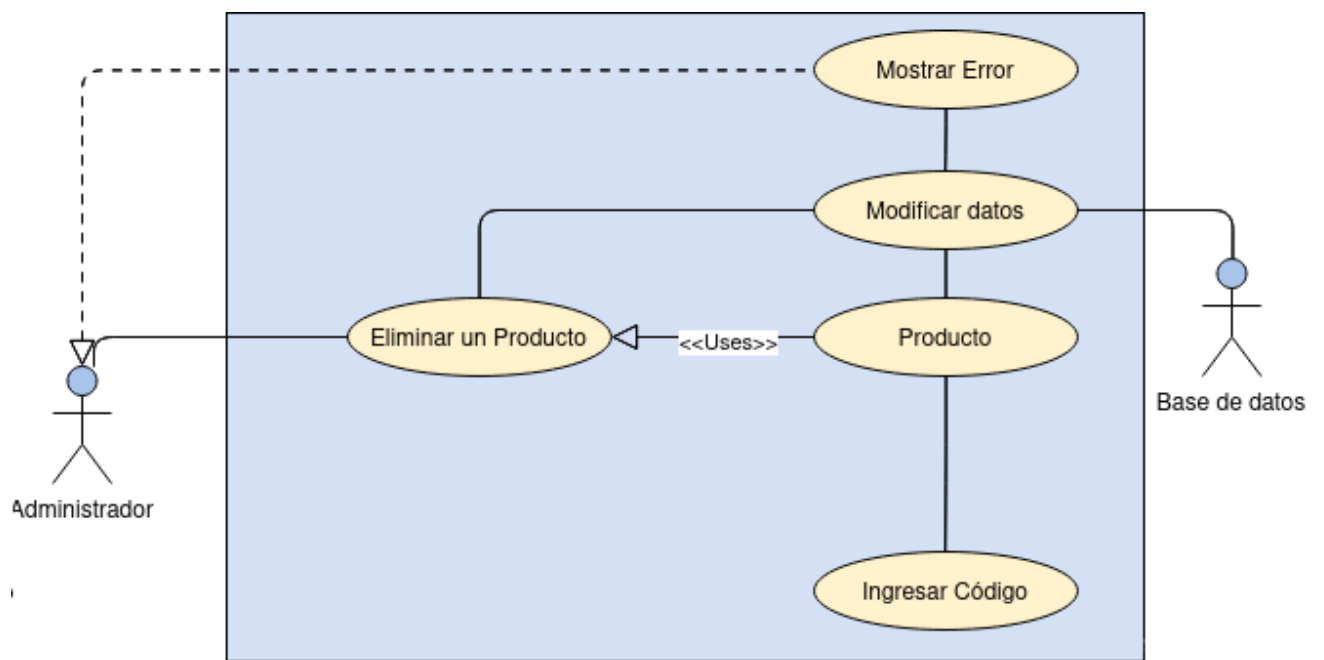


Figura 10.8: Diagrama de CU Eliminar Producto

ID:	CU-5		
Nombre:	Eliminar producto		
Creado por:	GJLA	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	01/12/2021
Actores:	Administrador, Usuario, Base de datos		
Descripción:	Elimina producto del sistema,		
Disparador:	Se selecciona la opción “Productos” del menú y enseguida presionar el botón “Eliminar” .		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se refresca la pantalla.		
Flujo normal:	1. Se ingresa código de producto que desea eliminar. 2. El sistema muestra en pantalla los datos nombre y descripción del producto que coincide con el código ingresado. 3. Usuario presiona botón “eliminar” y se hace borrado lógico del producto en la base de datos.		
Flujo alterno:	Se refresca la pantalla “Eliminar producto” , para que usuario pueda eliminar otro producto si lo desea.		
Includes:	N/A		
Frecuencia de uso:	Baja		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.9: CU Eliminar Producto

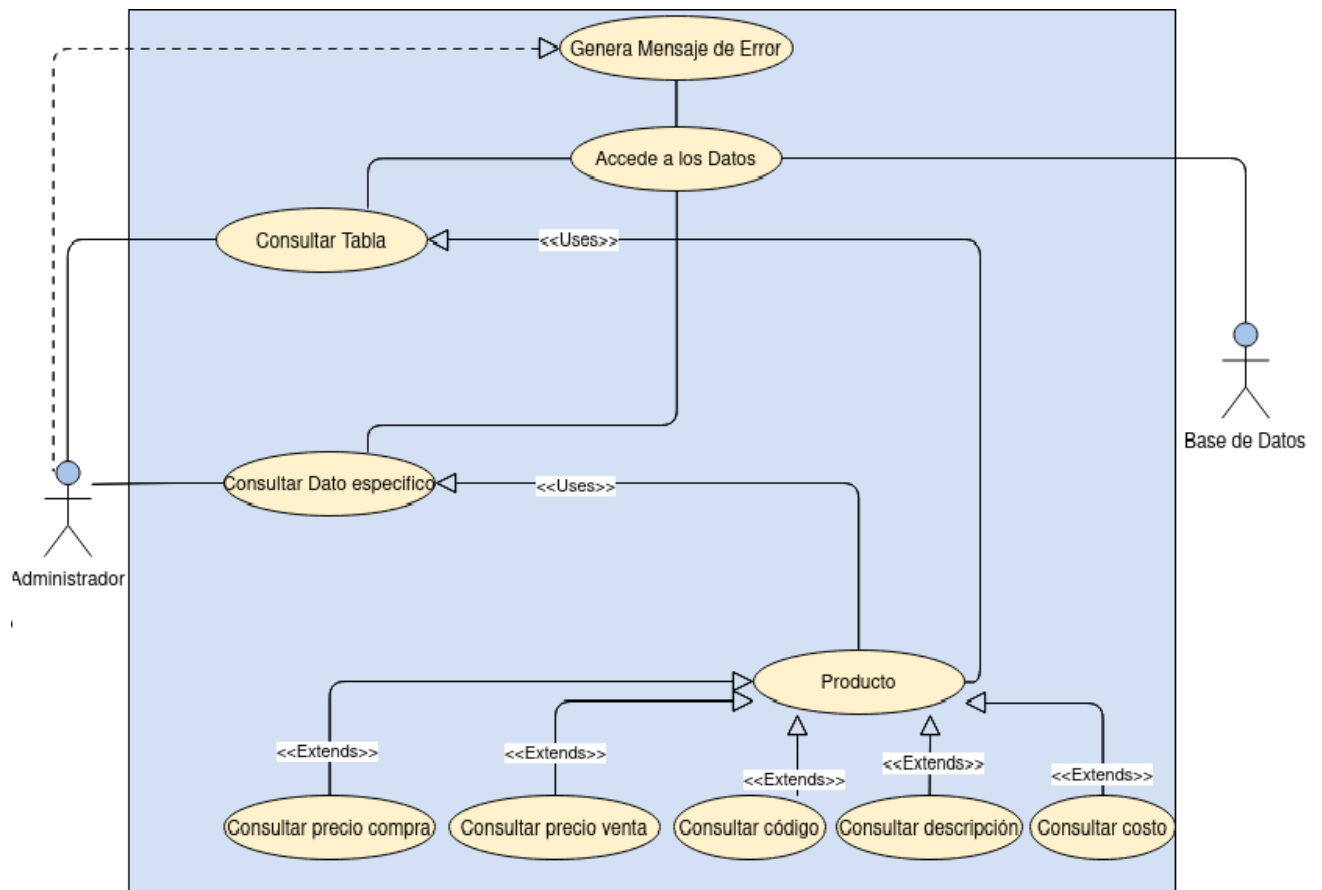


Figura 10.10: Diagrama de CU Inventario

ID:	CU-6		
Nombre:	Inventario		
Creado por:	MAG	Actualizado por:	MAG
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, usuario, base de datos.		
Descripción:	Muestra artículos registrados en el sistema.		
Disparador:	Se selecciona la opción “Inventario” del menú principal		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	N/A		
Flujo normal:	1. Al seleccionar opción “Inventario” del menú, el sistema muestra en una tabla los productos existentes en la base de datos. 2. Se ingresa código para ver datos de un producto en específico.		
Flujo alterno:	Usuario puede dirigirse a cualquier opción del menú.		
Includes:	N/A		
Frecuencia de uso:	Baja		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.11: CU Inventario

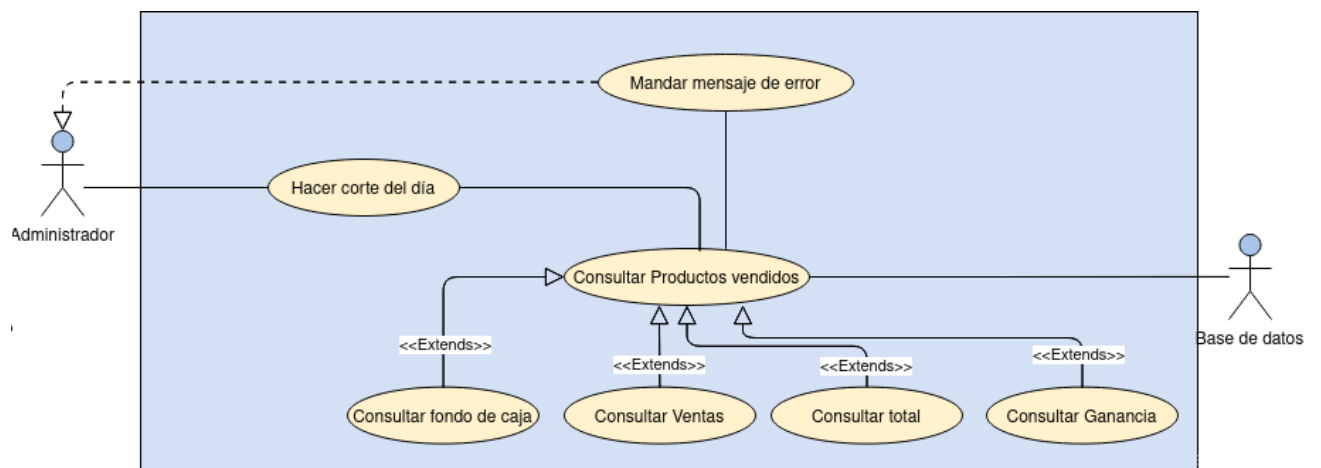


Figura 10.12: Diagrama de CU Corte

ID:	CU-7		
Nombre:	Generar corte de caja		
Creado por:	GJLA	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, usuario, base de datos.		
Descripción:	Genera corte de caja del día, mostrando datos como Fondo de caja, Ventas (importe total de las ventas del día), total (dinero total en caja) y Ganancia.		
Disparador:	Se selecciona la opción “Corte” del menú de opciones.		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	N/A		
Flujo normal:	1. Pulsa el botón “Hacer corte del día”. 2. El sistema muestra en una tabla la siguiente información: Fondo de caja, Ventas, Total y Ganancia.		
Flujo alterno:	Usuario puede dirigirse a cualquier opción del menú.		
Includes:	N/A		
Frecuencia de uso:	Frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.13: CU Corte

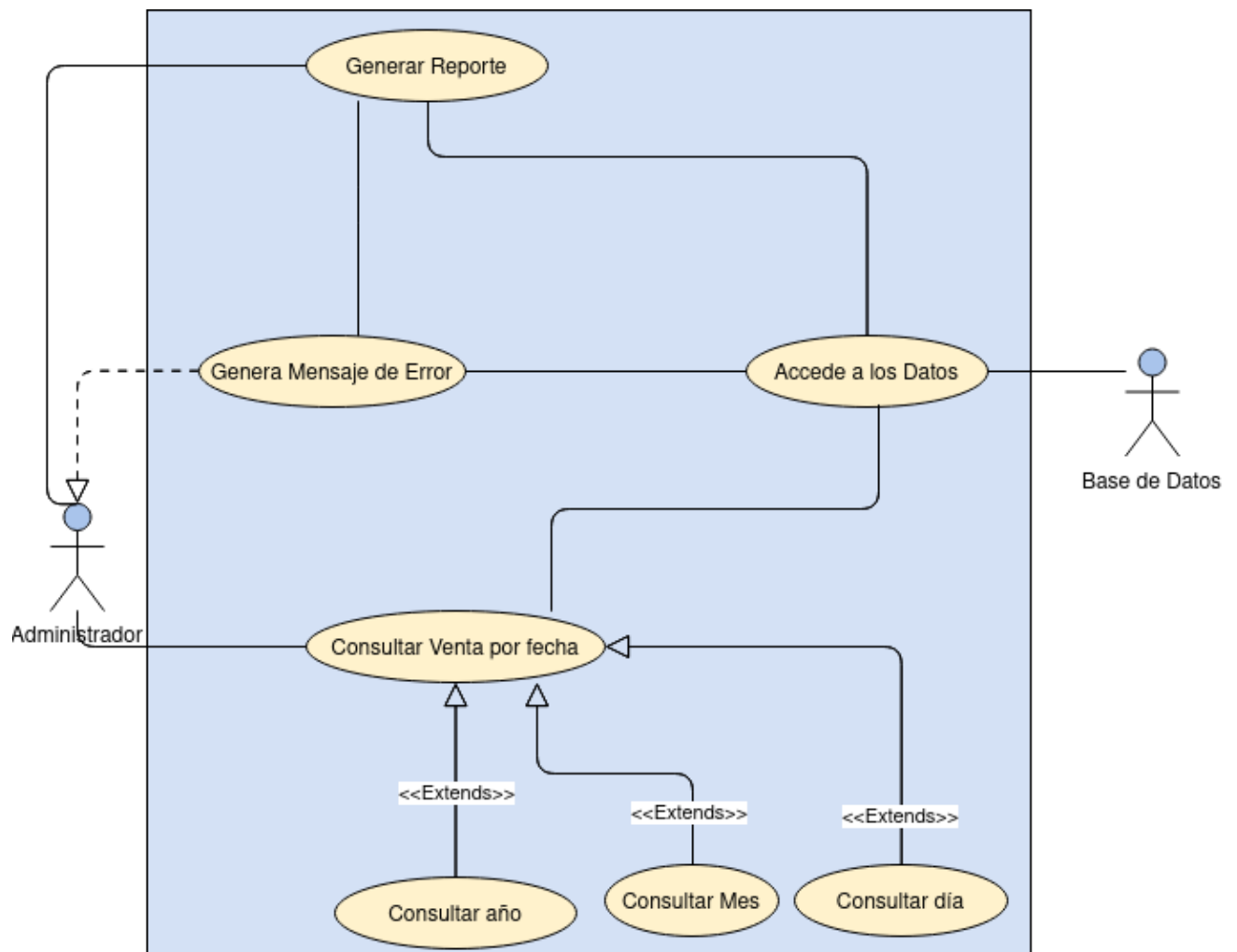


Figura 10.14: Diagrama de CU Reporte

ID:	CU-8		
Nombre:	Generar reporte		
Creado por:	RRR	Actualizado por:	RRR
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, base de datos.		
Descripción:	Genera reporte de ventas por día.		
Disparador:	Se selecciona la opción “Reporte” del menú de opciones.		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	N/A		
Flujo normal:	1. Se pulsa opción “Reporte” del menú de opciones y se muestra en una tabla el resumen de ventas por día de la semana, así como las ventas totales del día, el número de ventas realizadas así como el importe por venta promedio.		
Flujo alterno:	Usuario puede dirigirse a cualquier opción del menú.		
Includes:	N/A		
Frecuencia de uso:	Baja		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.15: CU Reporte

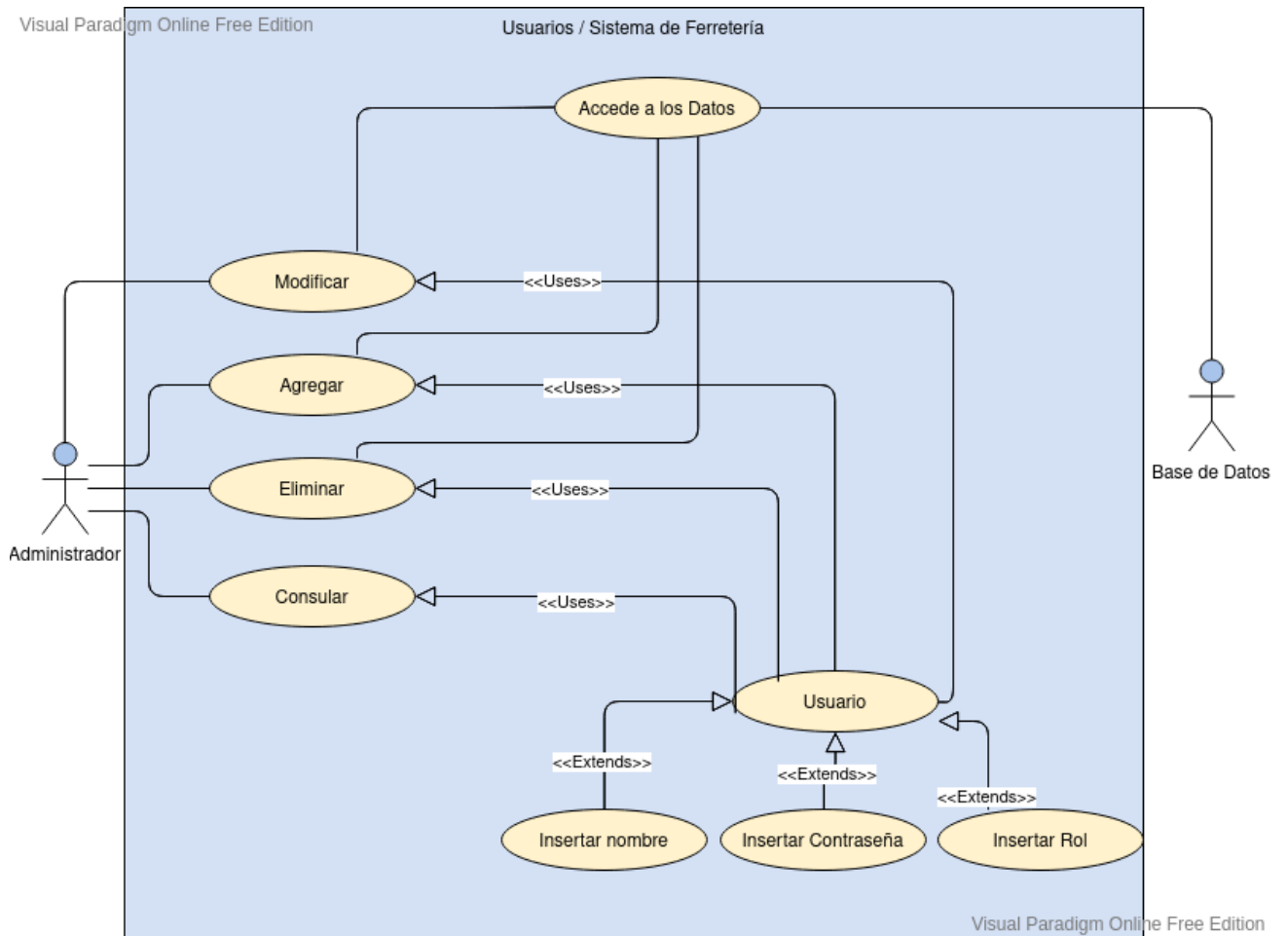


Figura 10.16: Diagrama de CU Usuarios

ID:	CU-9		
Nombre:	Agregar usuario		
Creado por:	RRR	Actualizado por:	RRR
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, base de datos.		
Descripción:	Registra nuevos usuarios en el sistema.		
Disparador:	Se selecciona la opción “Usuarios” del menú de opciones.		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se refresca pantalla, para que administrador pueda registrar un nuevo usuario si lo desea.		
Flujo normal:	1. Se ingresan datos del nuevo usuario a registrar: Nombre, Contraseña y rol (administrador o usuario). 2. Pulsa botón “Agregar usuario”. 3. Se registra nuevo usuario en la base de datos.		
Flujo alterno:	Se refresca la pantalla “Usuarios” , para que administrador pueda dar de alta a otro usuario si lo desea.		
Includes:	N/A		
Frecuencia de uso:	Muy frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.17: CU Usuario

Capítulo 11

Diccionario de datos

En la tabla 1 se muestran los tipos de datos correspondientes a los productos ya que al registrar un producto es necesario tener un código que lo identifique, un nombre con una breve descripción, sus correspondientes precios(venta y compra) para los futuros cálculos. Así como se debe llevar un control de la Existencia del producto y su marca.

Columna	Tipo de dato		Descripción
PK	idCodigo	idCod	Almacena el código único del producto
	nombre	Varchar(25)	Se guarda el nombre del producto
	descripcion	Varchar(50)	Se guarda una breve descripcion acerca de las cualidades del producto
	precioVenta	flotante	Almacena el precio por el cual se vende el producto correspondiente
	precioCompra	flotante	Almacena el precio por el cual se compra el producto correspondiente
	Existencia	flotante	Se guarda un total del total de productos correspondientes
	Marca	Varchar(30)	Se guarda la marca del producto

Figura 11.1: Productos

En la tabla 2 Se muestran los Datos que se ocuparan para las ventas realizadas, se debe contar con un identificador de venta para la generación de reportes a futuro, en la descripción de venta se registraran los productos que formaron parte de esta venta, el total de venta y la fecha en la que se realizó.

Columna		Tipo de dato	Descripción
PK	idVenta	idV	Identificador de la venta realizada
	descripcion	varchar(50)	Se registraran los productos adquiridos por el cliente así como el total de ellos
	totalVenta	flotante	Guarda el total de la venta realizada
	fecha	varchar(10)	Almacena la fecha en la cual se realizó la venta

Figura 11.2: Ventas

La Tabla 3 nos muestra las categorías en esta tabla se guardaran contando con un id de la categoría para evitar que se duplique, así como las distintas categorías (Equipamiento, cerrajería, fontanería, materiales eléctricos, herramientas, etc), cuenta además con una descripción breve de esta misma.

Columna		Tipo de dato	Descripción
PK	idCategoria	idCat	Identificador de la categoría del producto (Industrial, Hogar, etc)
	nombre	Integer	Almacena el nombre de la categoría
	descripcion	Integer	Se guarda una pequeña descripción del tipo de categoría que se hace referencia

Figura 11.3: Categorías

La Tabla 4 muestra los datos que se almacenaran de los empleados correspondientes, contando con su nombre, apellidos, cargo, y su propio identificador así como el identificador de la venta que realizó.

Columna		Tipo de dato	Descripción
PK	idEmpleado	idE	Identificador del empleado
FK	idVenta	idV	Identificador de la venta realizada
	nombre	varchar(25)	Nombre del empleado
	apellidos	varchar(25)	Apellidos del empleado
	cargo	varchar(15)	Cargo que ocupa el empleado

Empleado

La Tabla 5 funciona como enlace entre la tabla Productos y la tabla de Categoría esto para que se almacenen los productos con su categoría correspondiente.

Columna		Tipo de dato	Descripción
FK	idCodigo	idCod	Identificador del producto
FK	idCategoria	idCat	Identificador de la categoría a la que pertenece el producto

Pertenece

La Tabla 6 funciona como enlace entre la tabla Productos y la tabla de Ventas, esto para que se almacenen los ventas realizadas de los productos guardados correspondiente.

Columna		Tipo de dato	Descripción
FK	idCodigo	idCod	Identificador del producto
FK	idVenta	idV	Identificador de la venta realizada

Registro

Capítulo 12

Diagrama Conceptual

Este proyecto ha sido elaborado dada la necesidad de automatizar un proceso de ventas de una ferretería, la cual requiere generar informes exactos sobre las ventas diarias, además de que requiere de llevar un control de inventario de productos para poder supervisar el stock disponible de los mismos, además se pretende que a través de este sistema los empleados mejoren sus tiempos de atención al cliente en ventas ya que evitarían hacer cuentas de forma manual.

El sistema de software que implementaremos facilitará el manejo de ventas, inventario, ingreso de productos, etc. ya que esto se realizará de manera sistematizada dado que actualmente se realiza de manera manual, por lo anterior de este sistema ayudaría a visualizar el rendimiento de la empresa tomando en cuenta las ventas generadas por el sistema.

Este proyecto será realizado con el lenguaje de programación Java, además de que teniendo en cuenta que es un sistema que va a generar una gran cantidad de datos debido a las diferentes funcionalidades del mismo, se pretende hacer uso de una base de datos que los almacene. Como se mencionó antes este sistema contará con distintas funcionalidades las cuales ayudarían a agilizar tareas como la generación de reportes de ventas por días, generación de corte diario, manejo de ventas por cliente, cobro por ventas, y el manejo del inventario.

Capítulo 13

Diseño Lógico

El diagrama ER que se muestra a continuación representa todos los elementos que conforman a las entidades en la base de datos, así como sus respectivas llaves primarias y las relaciones que las entidades comparten siendo uno a uno, uno a muchos o muchos a muchos.

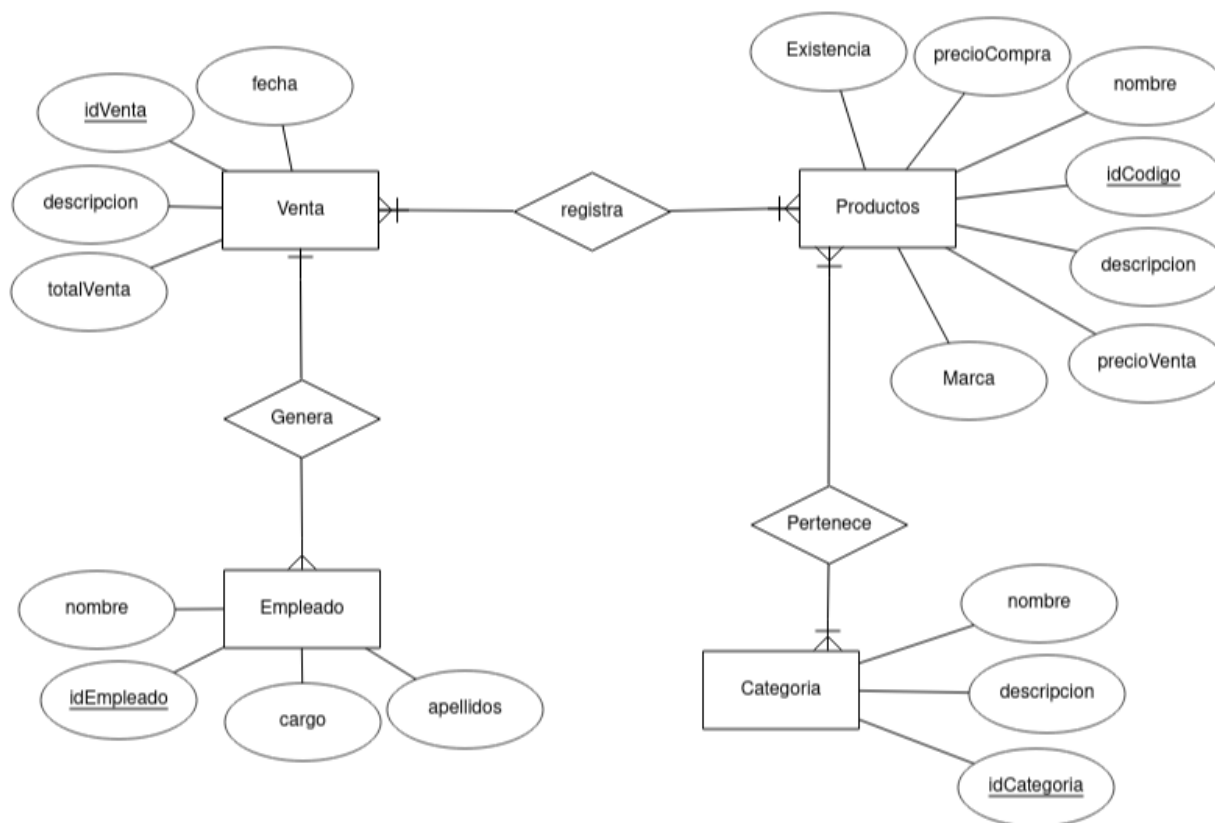


Figura 13.1: Diagrama ERD

El modelo Relacional que se muestra a continuación representa la base de datos, que corresponde al proyecto, en este modelo se puede observar el flujo que se necesita seguir cuando se consultan los valores de la base de datos.

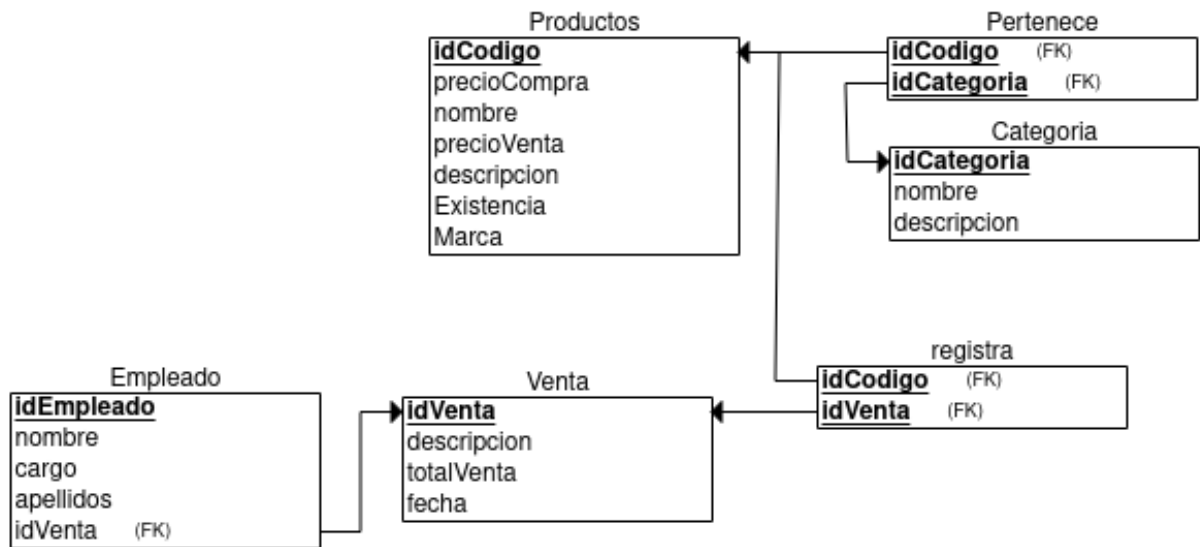


Figura 13.2: Esquema Relacional

Capítulo 14

Diseño Físico

```
create database ferregra;
\c ferregra

create domain flotante as text constraint flotante_check check (value ~'^[0-9]{1,9}.[0-9]{0,2}$'::text);
create domain idV as text constraint idV_check check (value ~'^V[0-9]{1,9}$'::text);
create domain idCod as text constraint idCod_check check (value ~'^PC[0-9]{1,9}$'::text);
create domain idCat as text constraint idCat_check check (value ~'^C[0-9]{1,9}$'::text);
create domain idE as text constraint idE_check check (value ~'^E[0-9]{1,9}$'::text);

create table Productos(idCodigo idCod primary key,
                      nombre varchar(25),
                      descripcion varchar(50),
                      precioVenta flotante,
                      precioCompra flotante,
                      Existencia integer,
                      Marca varchar(30));

create table Venta(idVenta idV primary key,
                  descripcion varchar(50),
                  totalVenta flotante,
                  fecha varchar(10));

create table Categoria(idCategoria idCat primary key,
                      nombre varchar(25),
                      descripcion varchar(50));

create table Empleado(idVenta idV,
                     idEmpleado idE primary key,
                     nombre varchar(25),
                     apellidos varchar(25),
                     cargo varchar(15),
                     FOREIGN KEY(idVenta)REFERENCES Venta(idVenta) ON DELETE CASCADE ON UPDATE CASCADE);

create table Pertenece(idCodigo idCod,
                      idCategoria idCat,
```

Figura 14.1: Script

```
create table Pertenece(idCodigo idCod,  
                      idCategoria idCat,  
                      FOREIGN KEY(idCodigo)REFERENCES Productos(idCodigo) ON DELETE CASCADE ON UPDATE CASCADE,  
                      FOREIGN KEY(idCategoria)REFERENCES Categoria(idCategoria) ON DELETE CASCADE ON UPDATE CASCADE);  
  
create table Registra(idCodigo idCod,  
                     idVenta idv,  
                     FOREIGN KEY(idCodigo)REFERENCES Productos(idCodigo) ON DELETE CASCADE ON UPDATE CASCADE,  
                     FOREIGN KEY(idVenta)REFERENCES Venta(idVenta) ON DELETE CASCADE ON UPDATE CASCADE);
```

Figura 14.2: Script1.1

Capítulo 15

Diagrama de Paquetes

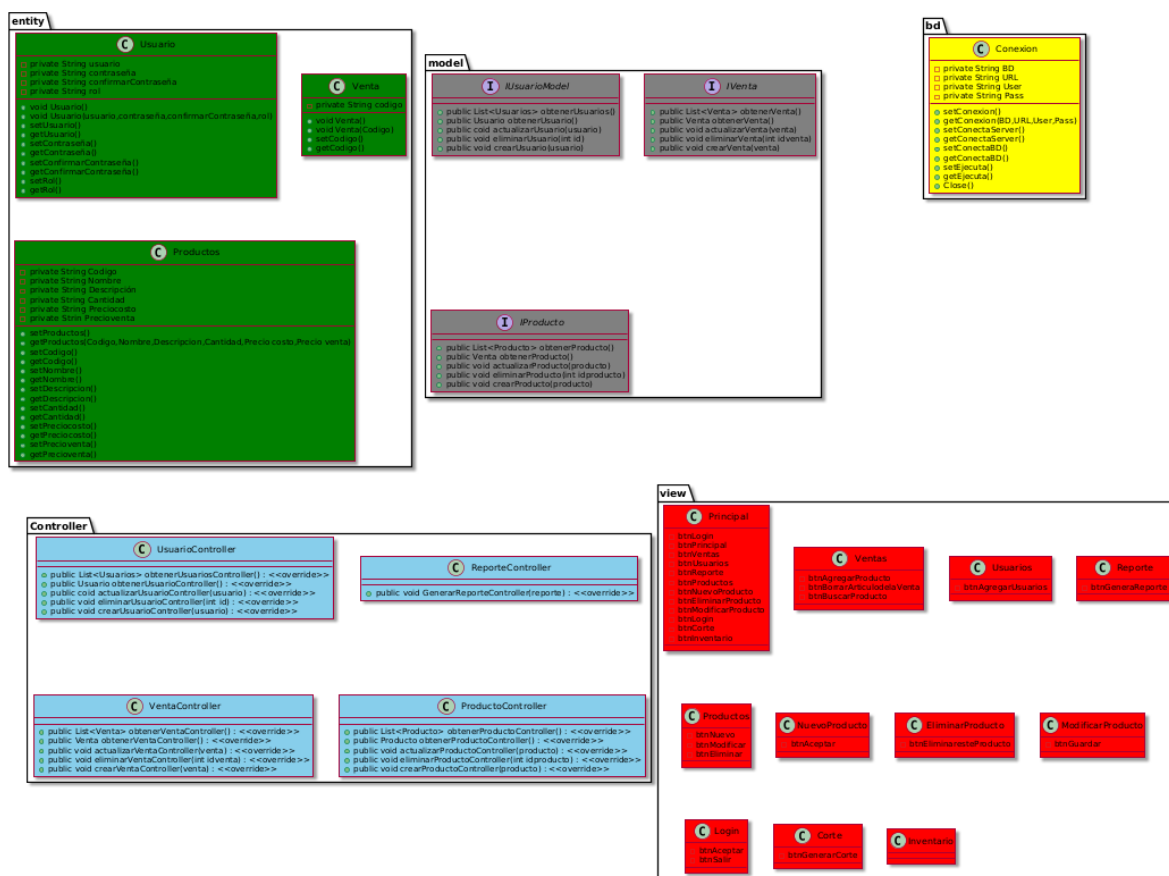


Figura 15.1: Diagrama de Paquetes

Capítulo 16

Diagrama de secuencias

Login

```
@startuml
Actor Administrador_Usuario
Administrador_Usuario -> Menu: Seleccionar opcion de menu

Administrador_Usuario <-- Menu: Opcion seleccionada
@enduml
```

Login1.1

```
@startuml
Actor Administrador
Actor Empleado
Administrador -> Administrar_datos: Ingresar datos
Activate Administrar_datos
Administrar_datos -> Administrar_datos: Usuario
Administrar_datos -> Administrar_datos: Contraseña
database Base_de_datos
Administrar_datos -> Base_de_datos: Validar informacion
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos-->Administrador: Acceso autorizado
deactivate Base_de_datos
@enduml
```

Nuevo Producto

```
@startuml
Actor Administrador
Administrador -> AgregarProducto
database BD
AgregarProducto-->AgregarDato
AgregarDato-->Producto
ModificarDato-->BD
Producto-->Error
ModificarDato-->Producto
```

```

AgregarDato-->Error
AgregarDato-->BD
@enduml

```

Consultar

```

@startuml
Actor Administrador
Administrador -> ConsultaDatos
Administrador -> ConsultaProductoEspecifico
database BD
ConsultaDatos-->BD
ConsultaProductoEspecifico->BD
BD-->Administrador
@enduml

```

Eliminar

```

@startuml
Actor Administrador
Administrador -> EliminarProducto
EliminarProducto-->ModificarDatos
database BD
ModificarDatos-->BD
Producto-->IngresarCódigo
ModificarDatos-->Error
ModificarDatos-->BD
BD-->Administrador
@enduml

```

Modificar Producto

```

@startuml
Actor Administrador
Administrador -> ModificarProducto
database BD
ModificarProducto-->ModificarDato
ModificarDato-->BD
ModificarDato-->Producto
ModificarDato-->Error
Producto-->Error
@enduml

```

Inventario

```

@startuml
Actor Administrador
Administrador -> ConsultarTabla
Administrador -> ConsultarDatoEspecifico
database BD
ConsultarTabla-->AccedeDatos
AccedeDatos-->BD
AccedeDatos-->MensajeError

```

```

Producto-->ConsultarDatoEspecifico
Producto-->ConsultarTabla
BD-->Administrador
@enduml

```

Corte

```

@startuml
Actor Administrador
Administrador -> HacerCorteDía
HacerCorteDía -> ConsultaProductosVendidos
database BD
ConsultaProductosVendidos-->BD
BD-->Administrador
@enduml

```

Reporte

```

@startuml
Actor Administrador
Administrador -> GeneraReporte
database BD
GeneraReporte-->AccedeDatos
AccedeDatos-->ConsultarVentaFecha
AccedeDatos-->BD
AccedeDatos-->Error
GeneraReporte-->Error
ConsultarVentaFecha-->ConsultarVentaFecha
ConsultarVentaFecha-->ConsultarVentaFecha
ConsultarVentaFecha-->ConsultarVentaFecha
@enduml

```

Ventas

```

@startuml
Actor Administrador
database BD
Administrador -> BuscarProducto
Administrador -> AgregarProductoVenta
Administrador -> BorrarrProductoVenta
Producto-->BD
ConsultarNombre-->Producto
ConsultarCodigo-->Producto
ConsultarDescripción-->Producto
ConsultarPrecioVenta-->Producto
ConsultarPrecioCosto-->Producto
ConsultarExistencia-->Producto
@enduml

```

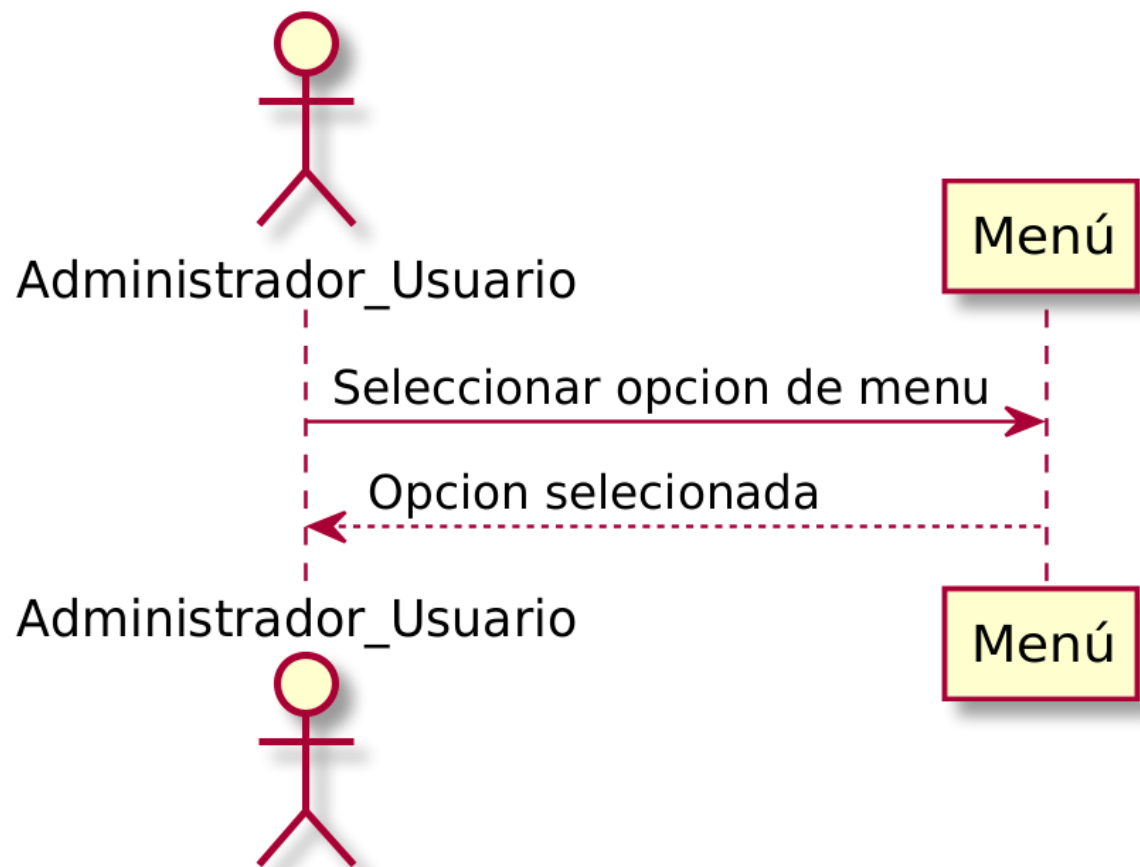


Figura 16.1: Login

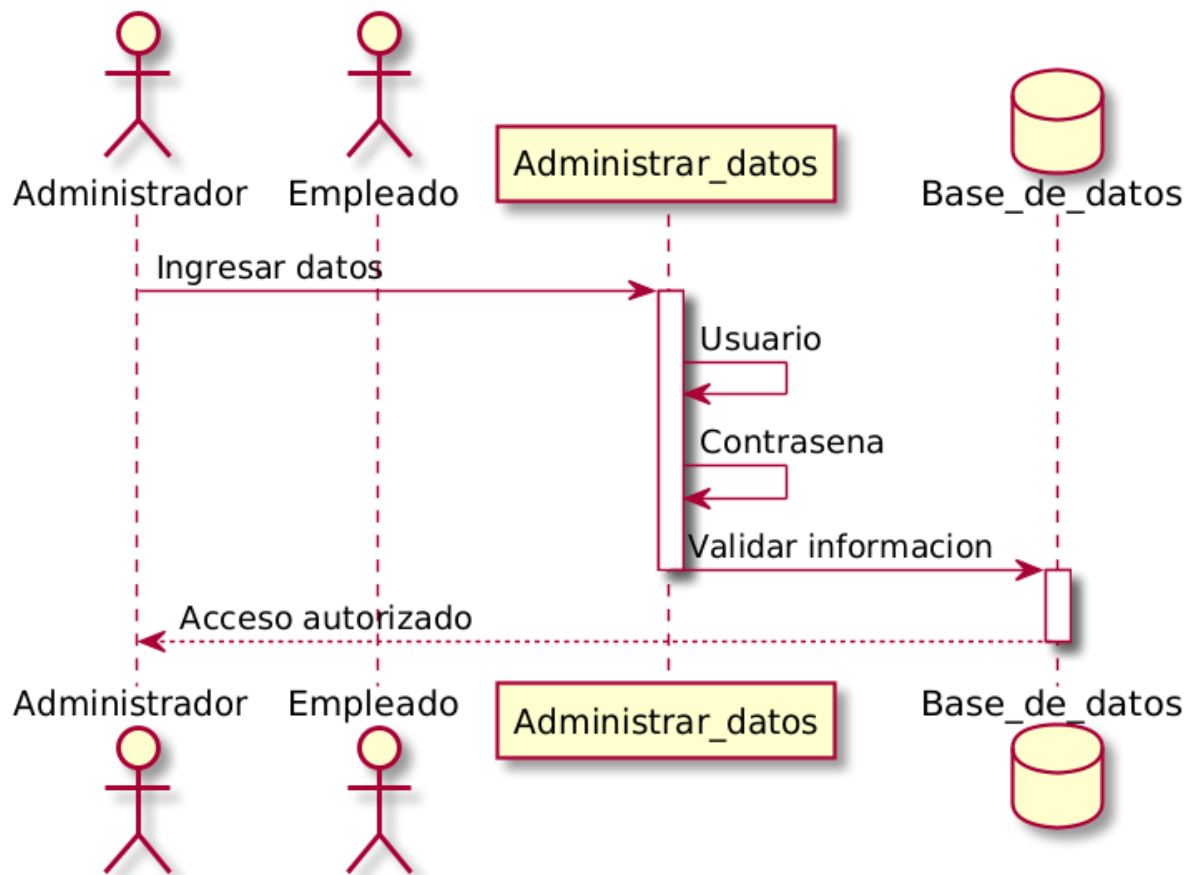


Figura 16.2: Login1.1

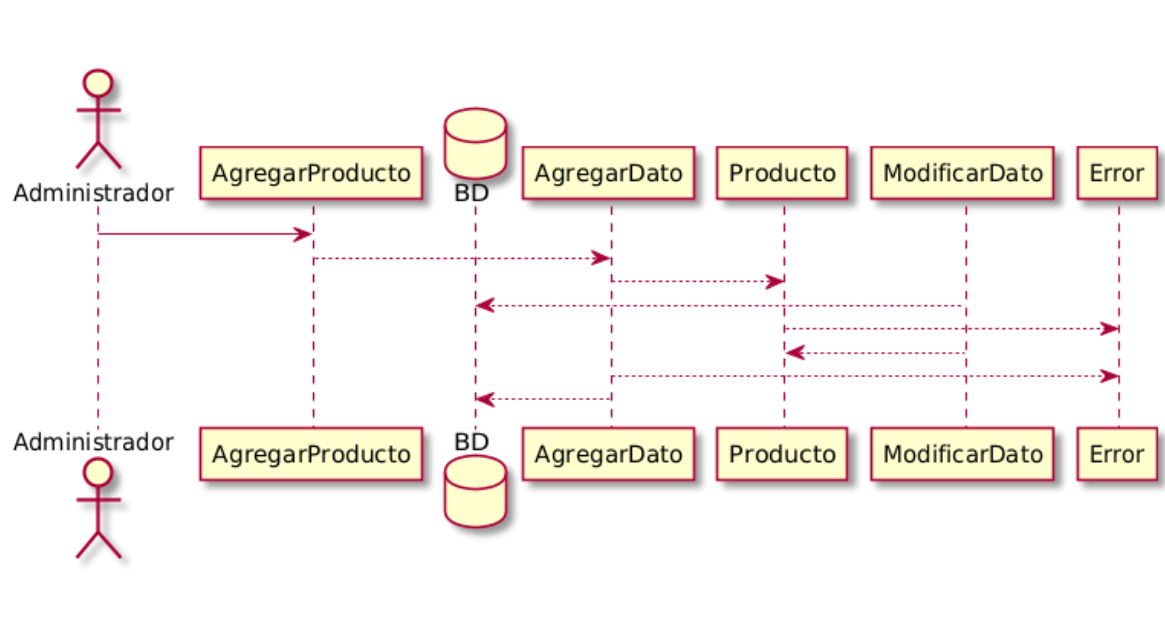


Figura 16.3: Nuevo Producto

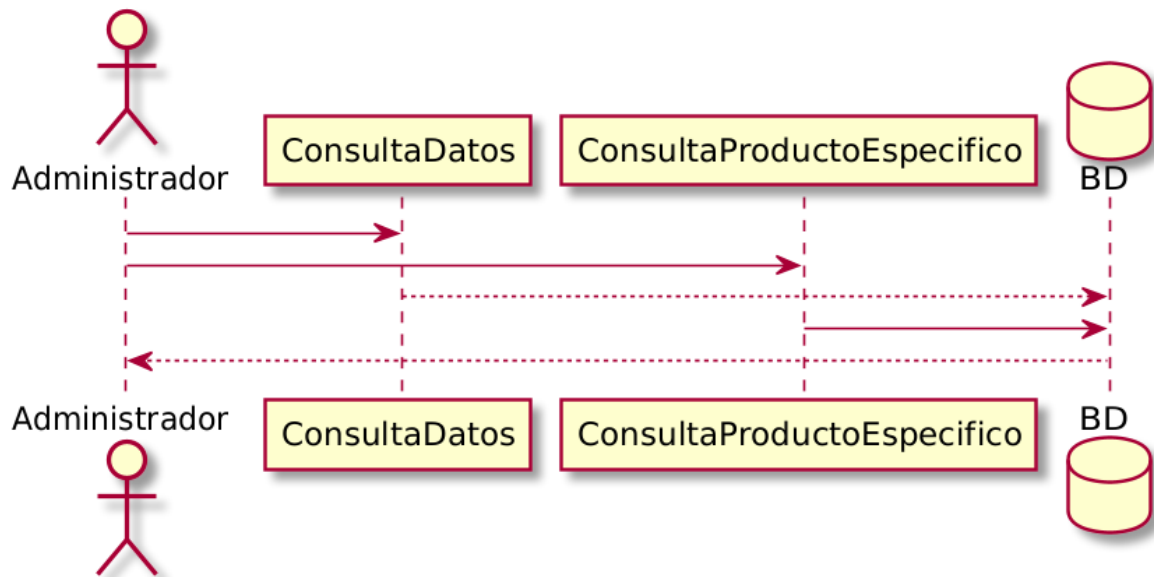


Figura 16.4: Consultar

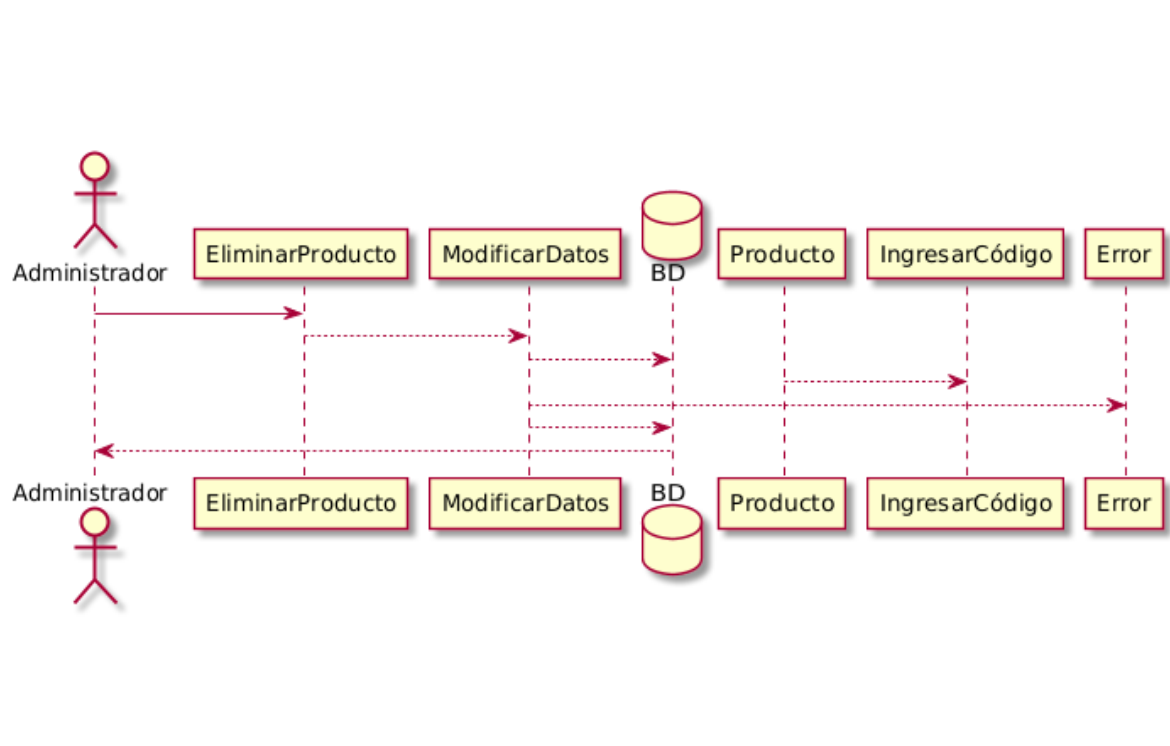


Figura 16.5: Eliminar

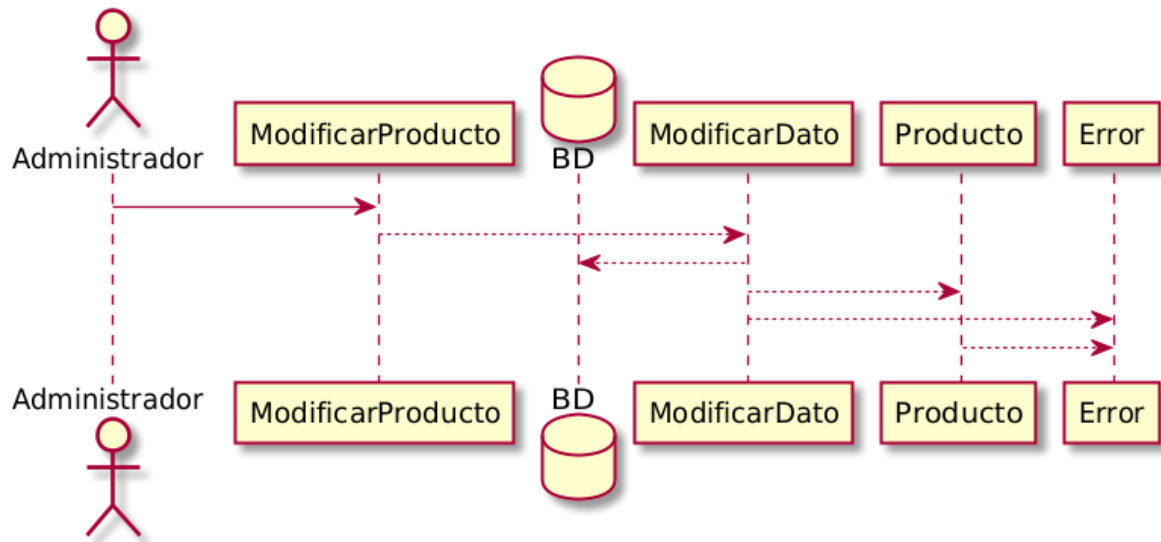


Figura 16.6: Modificar Producto

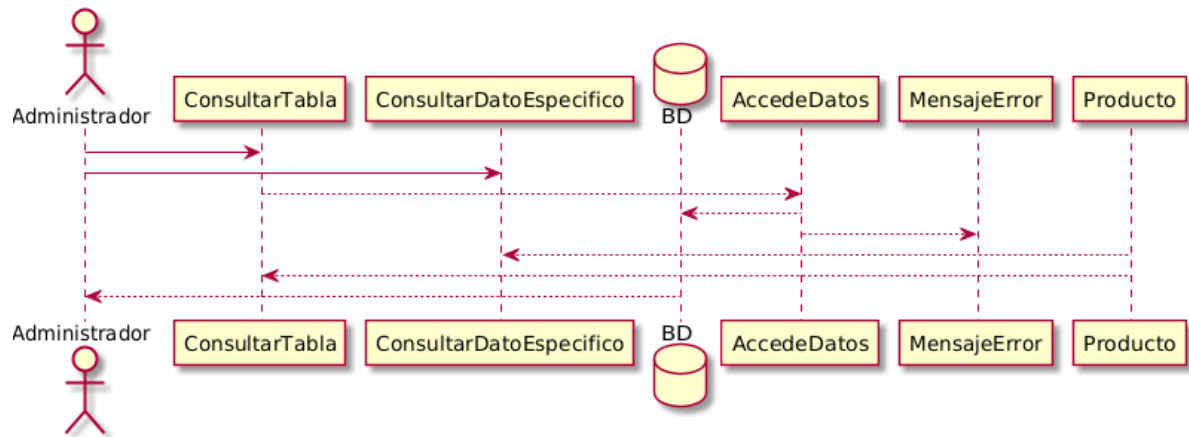


Figura 16.7: Inventario

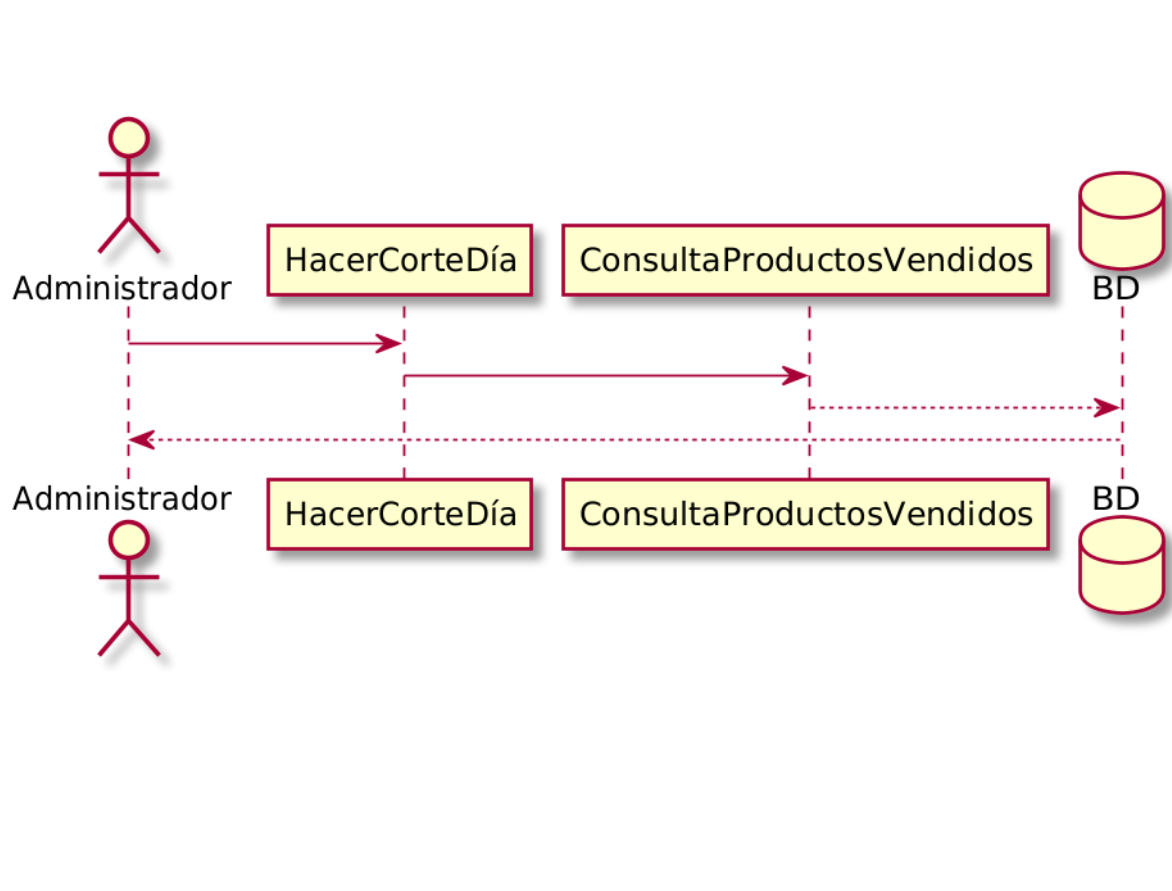


Figura 16.8: Corte

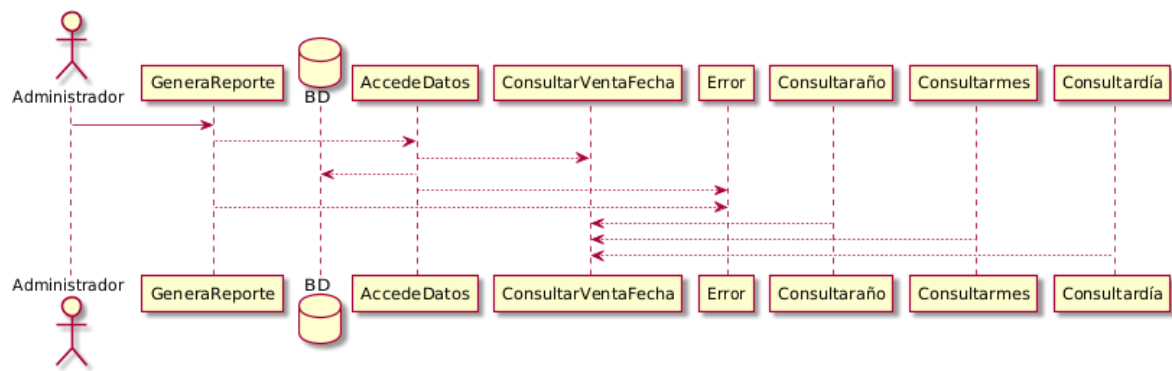


Figura 16.9: Reporte

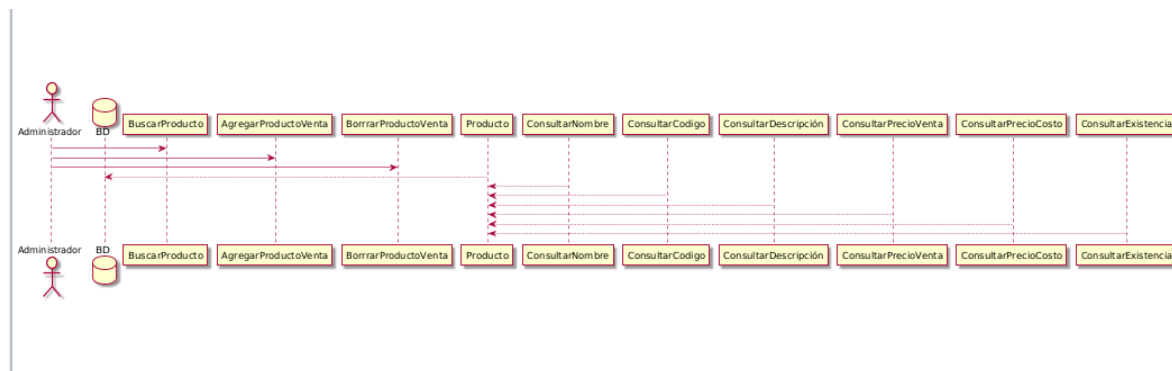


Figura 16.10: Ventas

Capítulo 17

Diagramas de Estado

Login

```
@startuml
[*] --> Usuario
[*] --> Administrador
Usuario :El usuario desea acceder
Administrador : Se solicita acceso al administrador
VerificaRegistro : Se registra el rol, contraseña.
Accesomenú: Acceso correctamente
Error: Muestra error(usuario/contraseña/ambos)
Usuario --> VerificaRegistro
Administrador --> VerificaRegistro
VerificaRegistro -->Error:manda error
Error-->VerificaRegistro
VerificaRegistro -->Accesomenú: inicia sesión
Accesomenú --> [*]
@enduml
```

Nuevo

```
@startuml
Administrador:Acceso Directo
AgregarProducto : Se agrega un nuevo producto
Producto : Se registra nombre, codigo, precios, existencia
Error: Muestra un error
AgregarDato:Se Agrega un dato a la BD
[*]--> Administrador
Administrador -->AgregarProducto
AgregarProducto --> Producto
Producto-->AgregarProducto
Producto--> Error:Manda Error
Error-->Producto
AgregarProducto--> AgregarDato
AgregarDato-->[*]
@enduml
```

Modificar

```

@startuml
[*] --> Administrador
Administrador : Acceso directo
Producto: ConsultarPrecioCompra,ConsultarCódigo,ConsultarDescripción,ConsultarPrecioVenta,ModificarPr
Administrador -->ConsultarTabla
Administrador -->ConsultarDatoEspecifico
ConsultarDatoEspecifico-->Producto
ConsultarTabla-->AccedeDatos
ConsultarTabla-->Producto: Accede a los productos
AccedeDatos-->ConsultarDatoEspecifico
AccedeDatos-->Error
ConsultarDatoEspecifico-->Error
Producto-->[*]
@enduml

```

Eliminar

```

@startuml
[*] --> Administrador
Administrador : Acceso directo
Administrador -->EliminarProducto
EliminarProducto-->ModificarDatos
Producto-->IngresarCódigo
ModificarDatos-->Producto
MostrarError-->ModificarDatos
MostrarError-->Producto
MostrarError-->IngresarCódigo
EliminarProducto-->[*]
@enduml

```

Consultar

```

@startuml
[*] --> Administrador
Administrador : Acceso directo
Producto: ConsultarNombre,ConsultarCódigo,ConsultarDescripción,ConsultarPrecioVenta,ConsultarPrecioCo
Administrador-->ConsultarDatos
ConsultarDatos-->Producto
Administrador-->ConsultarProductoEspecifico
ConsultarProductoEspecifico-->Producto
ConsultarProductoEspecifico-->Error
Producto-->Error
ConsultarDatos-->Error
ConsultarDatos-->[*]
@enduml

```

Inventario

```

@startuml
[*] --> Administrador
Administrador : Acceso directo
Producto: ConsultarPrecioCompra,ConsultarCódigo,ConsultarDescripción,ConsultarPrecioVenta,ModificarPr

```

```

Administrador -->ConsultarTabla
Administrador -->ConsultarDatoEspecifico
ConsultarDatoEspecifico-->Producto
ConsultarTabla-->AccedeDatos
ConsultarTabla-->Producto: Accede a los productos
AccedeDatos-->ConsultarDatoEspecifico
AccedeDatos-->Error
ConsultarDatoEspecifico-->Error
Producto-->[*]
@enduml

```

Corte

```

@startuml
@startuml
[*] --> Administrador
Administrador : Acceso directo
ConsultarProductoVendidos: ConsultarFondoCaja,ConsultarVentas,ConsultarTotal,ConsultarGanancia
Administrador-->CorteDía
CorteDía-->ConsultarProductoVendidos
Administrador-->ConsultarProductoVendidos
ConsultarProductoVendidos-->Error
CorteDía-->[*]
@enduml

```

Reporte

```

@startuml
[*] --> Administrador
Administrador : Acceso directo
ConsultarVentaPorFecha:Consultar año, Consultar mes, Consultar día
Error:
Administrador --> ConsultarVentaPorFecha:Se ingresan datos de las ventas
ConsultarVentaPorFecha-->AccesoDatos
AccesoDatos-->GeneraReporte
AccesoDatos-->Error
GeneraReporte-->Error
Error-->ConsultarVentaPorFecha
GeneraReporte-->[*]
@enduml

```

Ventas

```

@startuml
[*] --> Administrador
Administrador : Acceso directo
BuscarProducto :Nombre,Código,Descripción,PrecioVenta,PrecioCosto,Existencia
Error: Muestra error(usuario/contraseña/ambos)
Administrador --> BuscarProducto:Se ingresan datos del producto
Administrador--> BorrarProductoVenta: Producto inexistente
Administrador -->Error: Datos incorrectos
@enduml

```

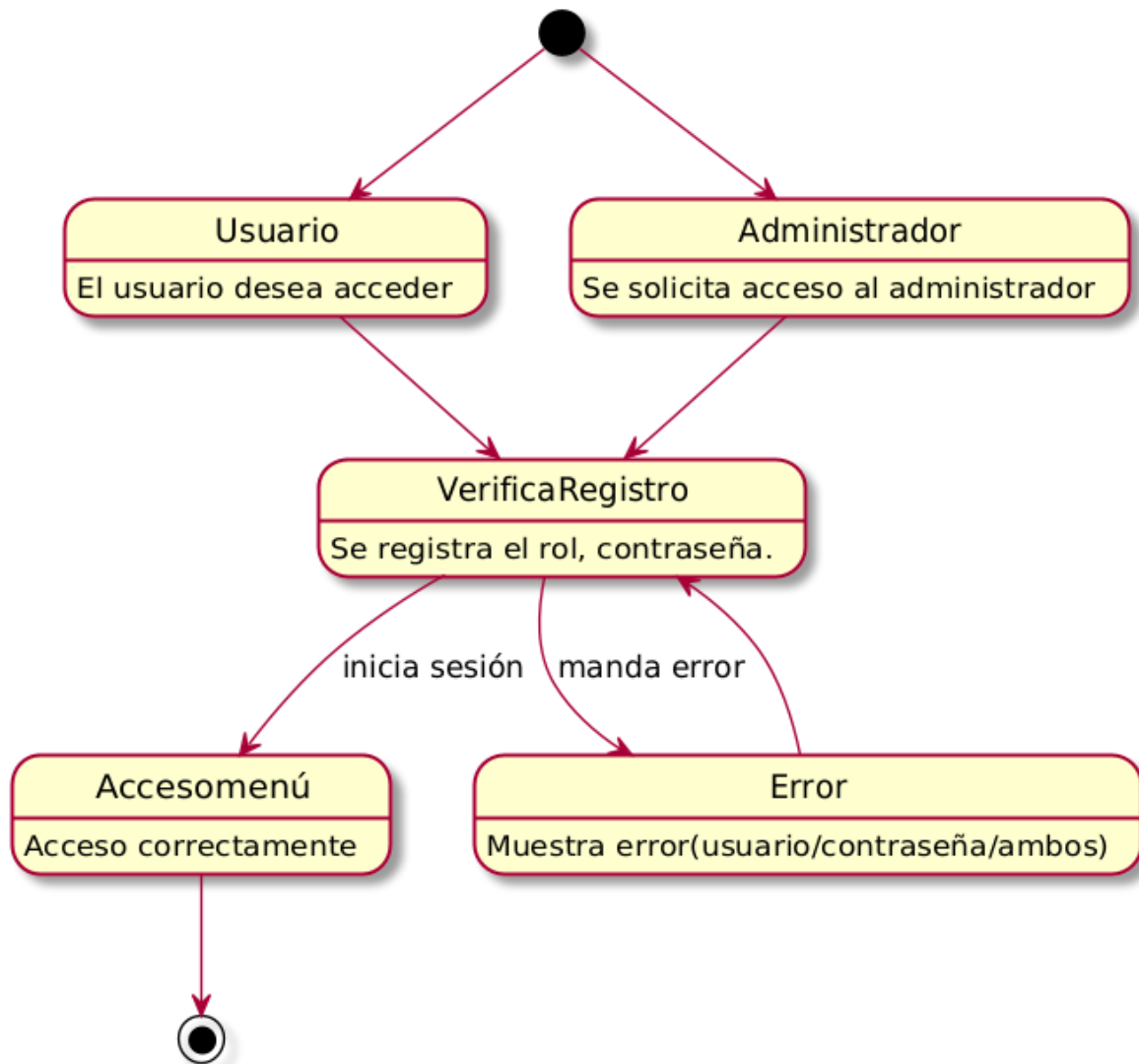


Figura 17.1: Login

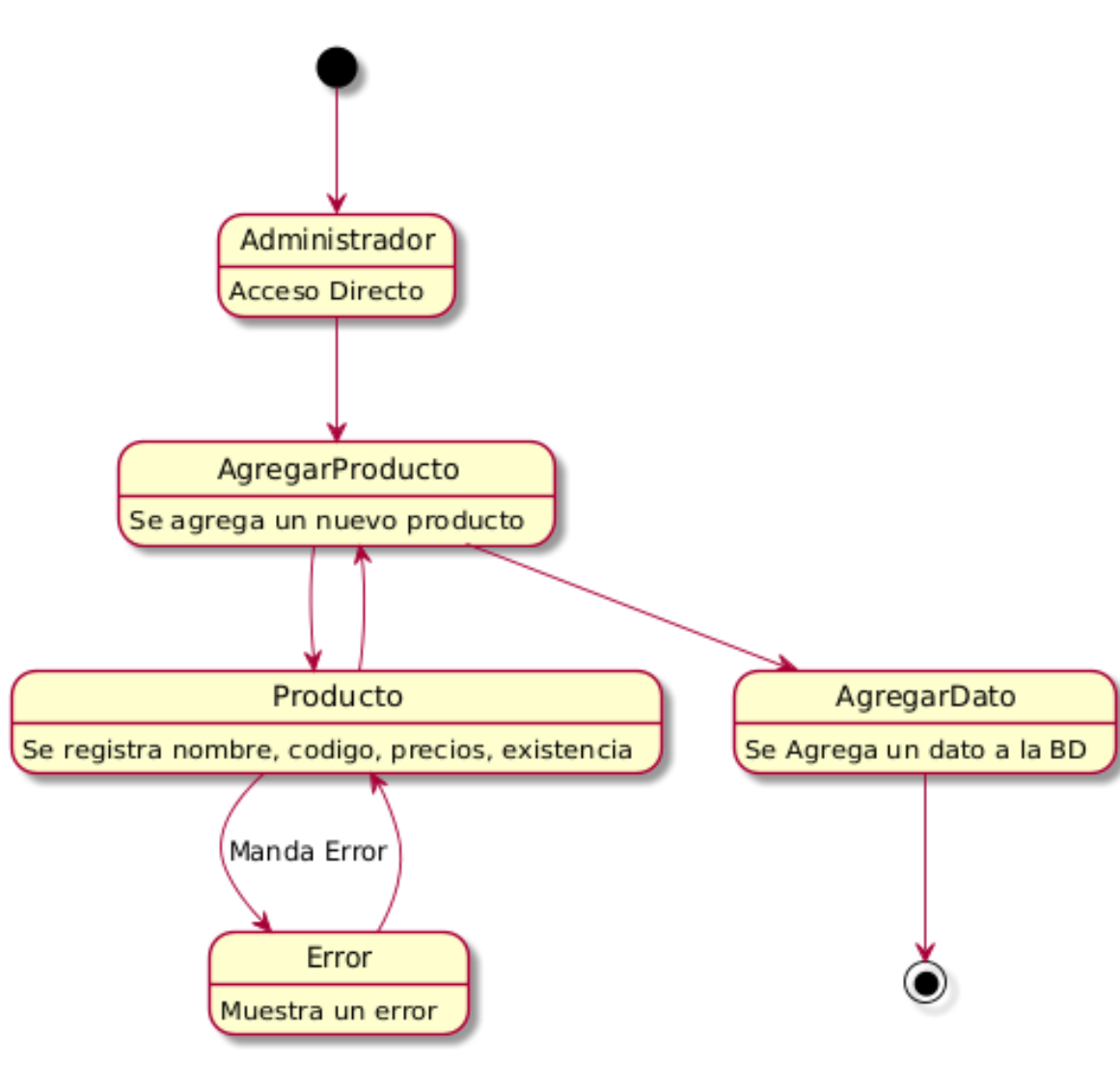


Figura 17.2: Nuevo

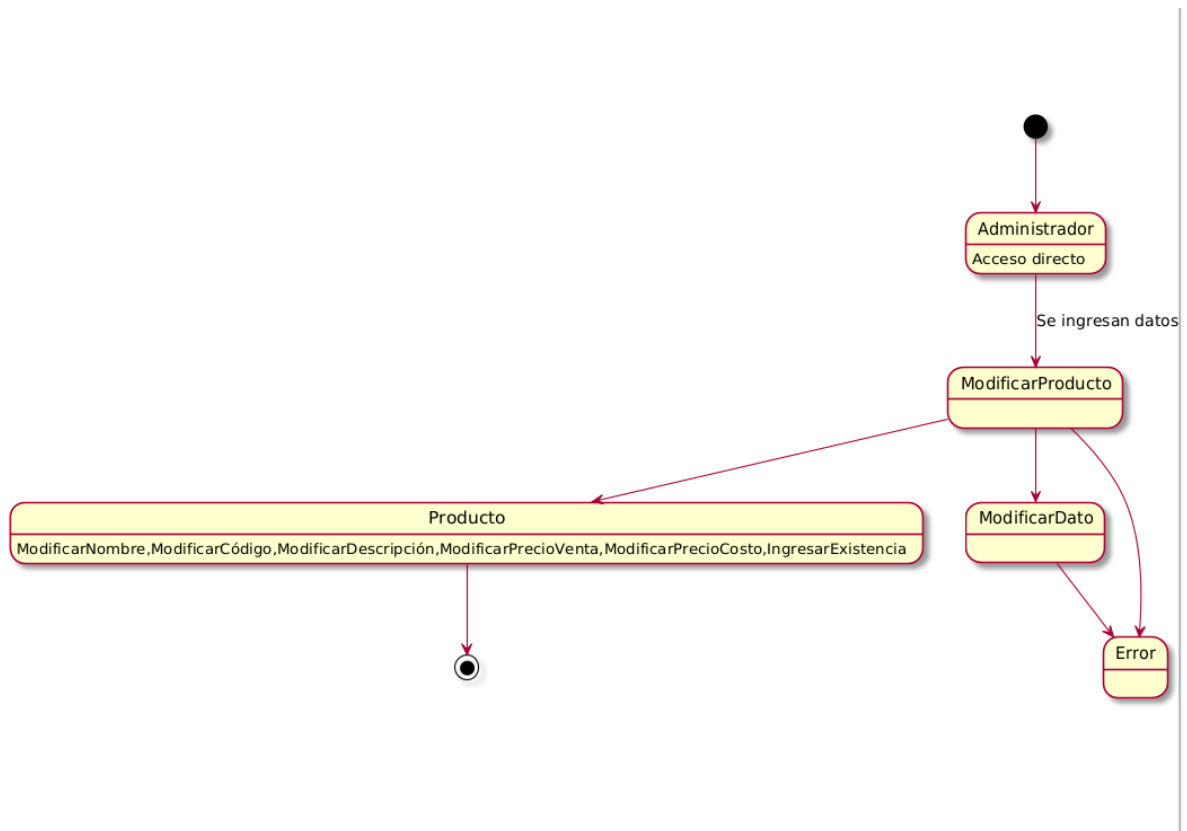


Figura 17.3: Modificar

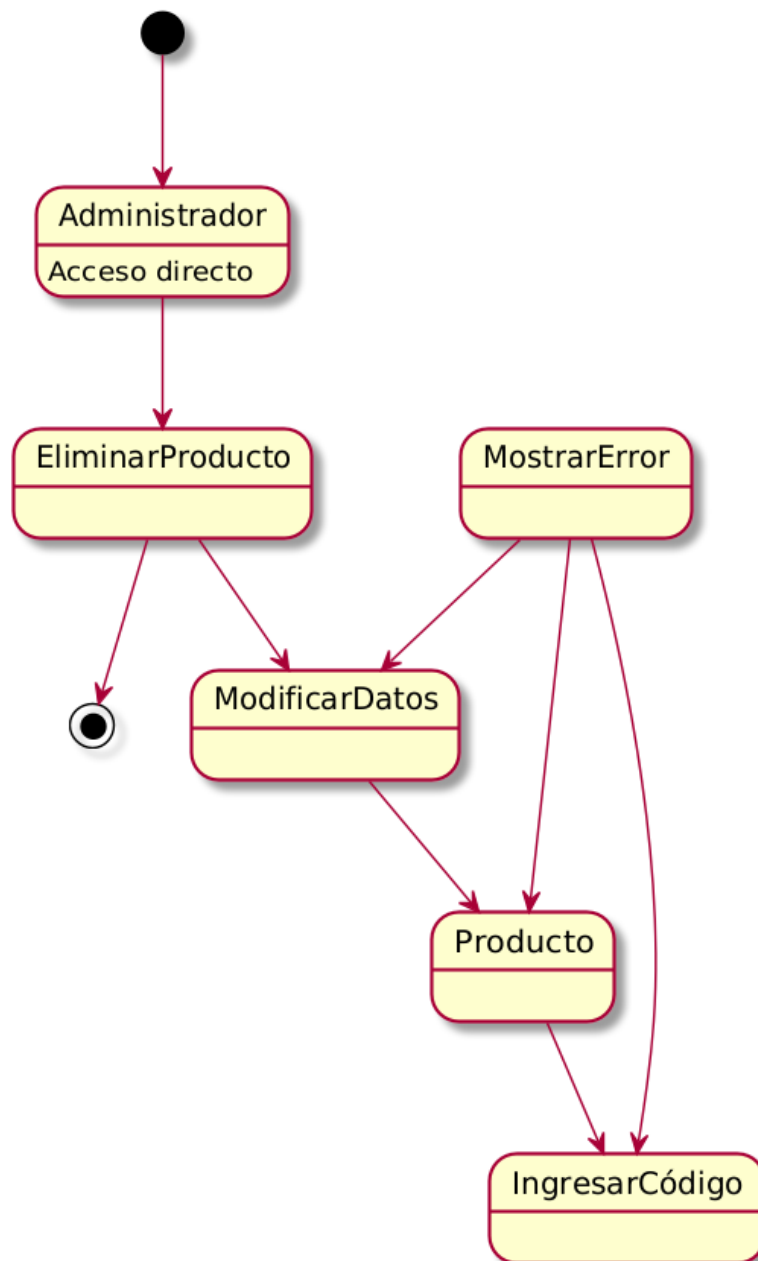


Figura 17.4: Eliminar

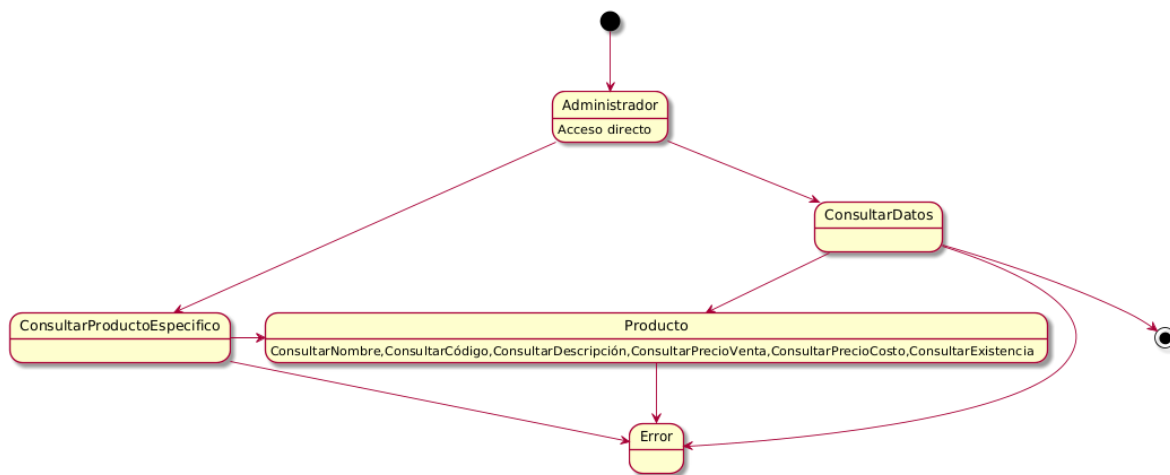


Figura 17.5: Consultar

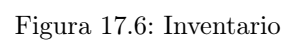


Figura 17.6: Inventario

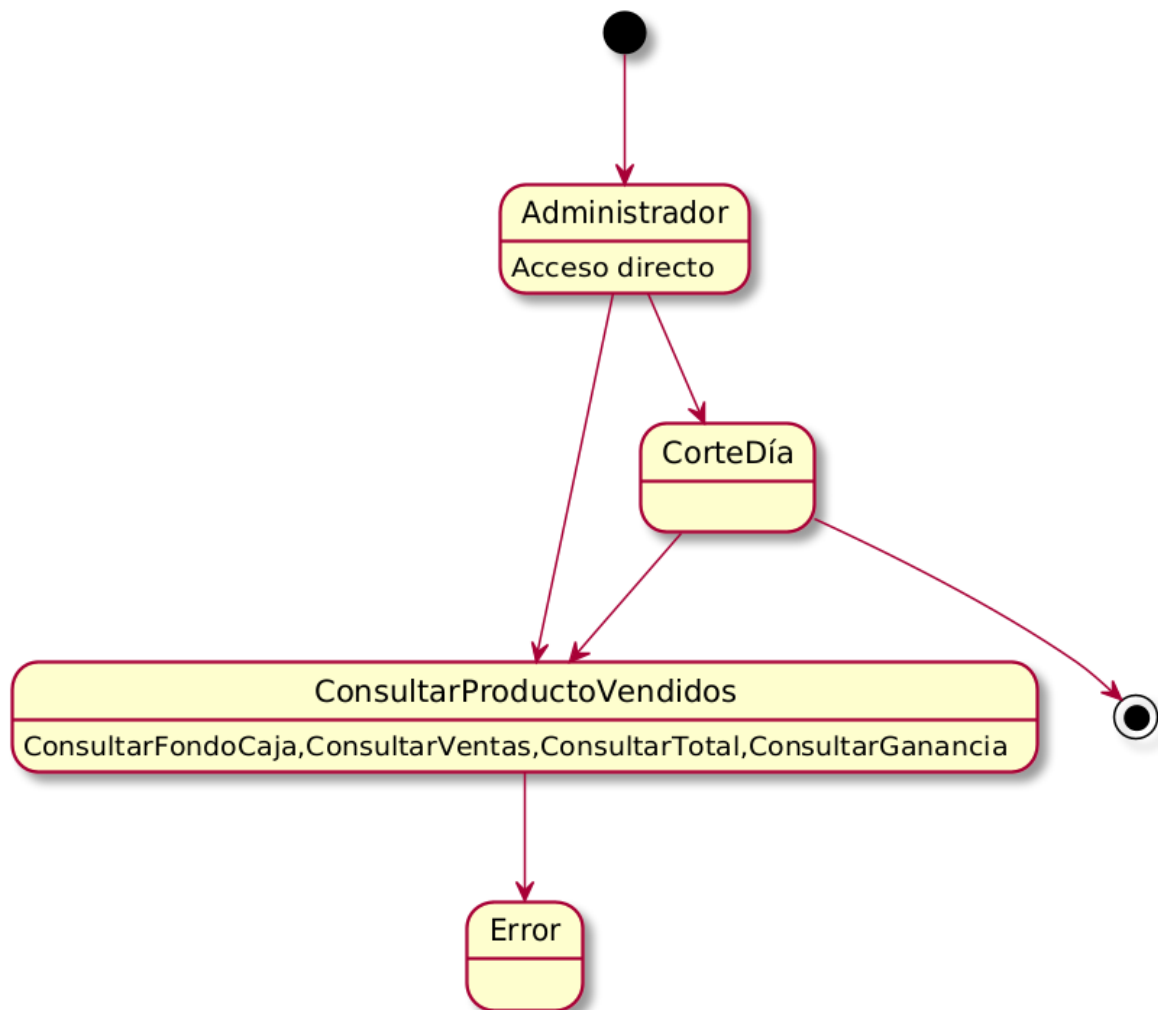


Figura 17.7: Corte

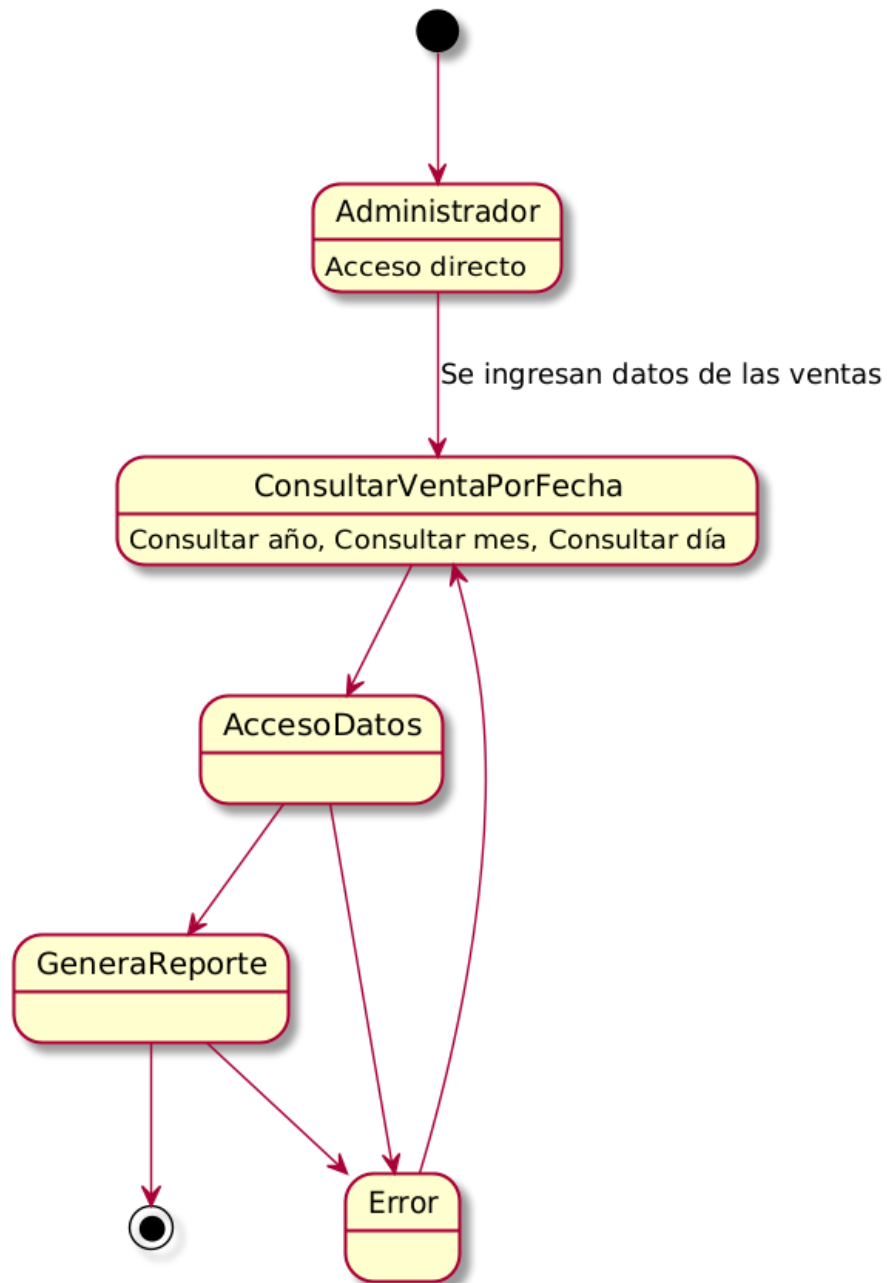


Figura 17.8: Reporte

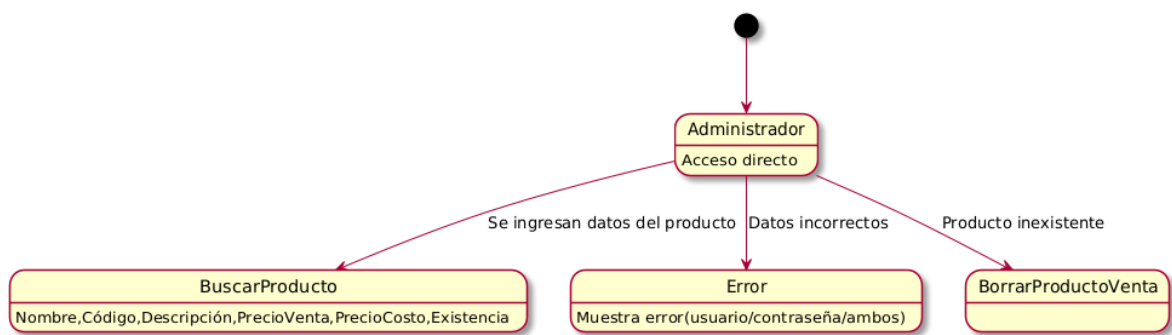


Figura 17.9: Ventas

Capítulo 18

Diagramas de Actividades

Login

```
@startuml
|Usuario|
start
    fork
        :Ingresar usuario;
        :Ingresar contraseña;
    |Administrador|
        fork again
            :Ingresar usuario;
            :Ingresar contraseña;
        endfork
    |Empleado|
        :Ingresar usuario;
        :Ingresar contraseña;
    |Base_de_datos|
        if (datos incorrectos) then (yes)
            :introducir datos correctos;
        else (no)
            :mensaje de datos no encontrados;
        endif
    stop
@enduml
```

Nuevo

```
|Administrador|
start

:Modificar Producto;
:Modificar Código;
:Modificar Descripción;
:Modificar Precio Venta;
:Modificar Precio Costo;
:Modificar Existencia ;
```



```
|Base_de_datos|
```

```
    if (datos correctos) then (yes)
        :Agregar nuevo producto;
    else (no)
        :mandar mensaje de error;
    endif
stop
```

Modificar

```
@startuml
```

```
|Administrador|
```

```
start
```

```
    :Modificar Nombre;
    :Modificar Código;
    :Modificar Descripción;
    :Modificar Precio Venta;
    :Modificar Precio Costo;
    :Ingresar Existencia;
```

```
|Base_de_datos|
```

```
    if (Producto encontrado) then (yes)
        :modificar datos;
    else (no)
        :mandar mensaje de Error;
    endif
```

```
|Administrador|
```

```
stop
```

```
@enduml
```

Eliminar

```
@startuml
```

```
|Administrador|
```

```
start
```

```
    :Ingresar código;
```

```
|Base_de_datos|
```

```
    if (Producto) then (yes)
        :elimina el producto;
    else (no)
        :mandar mensaje de error, modifica los datos;
    endif
```

```
|Administrador|
```

```
stop
```

```
@enduml
```

Consultar

```
@startuml
```

```

|Administrador|
start
  :Consultar Producto Específico;
  :Consultar Datos;
  :Consultar Código;
  :Consultar Descripción;
  :Modificar Precio Venta;
  :Modificar Precio Costo;
  :Consultar Existencia;

|Base_de_datos|
  if (Producto) then (yes)
    :modificar datos;
  else (no)
    :mandar mensaje de Error;
  endif
|Administrador|
stop
@enduml

```

Inventario

```

@startuml
|Administrador|
start
  :Consultar Dato Específico;
  :Consultar Tabla;
  :Producto;
  :Consultar precio compra;
  :Consultar precio venta;
  :Consultar código;
  :Consultar descripción;
  :Consultar costo;

|Base_de_datos|
  if (Producto) then (yes)
    :Accede a los datos;
  else (no)
    :mandar mensaje de error;
  endif
|Administrador|
stop
@enduml

```

Corte

```

@startuml
|Administrador|
start
  :Hacer corte del día;
  :Consultar Productos vendidos;

```

```

:Consultar fondo de caja;
:Consultar ventas;
:Consultar total;
:Consultar ganancia;

|Base_de_datos|
  if (consulta datos) then (yes)
    :Hacer corte;
  else (no)
    :mandar mensaje de Error;
  endif
|Administrador|
stop
@enduml

```

Reporte

```

@startuml
|Administrador|
start
  :Consultar Venta por fecha;
  :Consultar año;
  :Consultar mes;
  :Consultar día;

|Base_de_datos|
  if (Accede a los datos) then (yes)
    :Genera reporte;
  else (no)
    :mandar mensaje de error;
  endif
|Administrador|
stop
@enduml

```

Ventas

```

@startuml
|Administrador|
start
  :Buscar producto;
  :Producto;
  :Consultar Nombre;
  :Consultar Código;
  :Consultar Descripción;
  :Consultar Precio Venta;
  :Consultar Precio Costo;
  :Consultar Existencia;

|Base_de_datos|
  if (Accede a los datos(Producto)) then (yes)

```

```

        :Agrega producto a la venta;
    else (no)
        :mandar mensaje de error;
    endif
|Administrador|
stop
@enduml

```

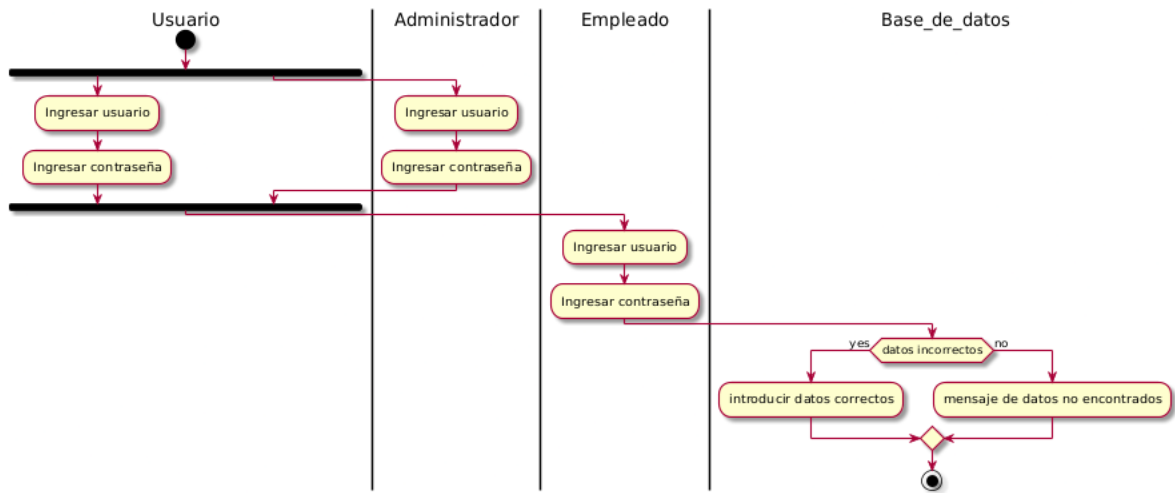


Figura 18.1: Login

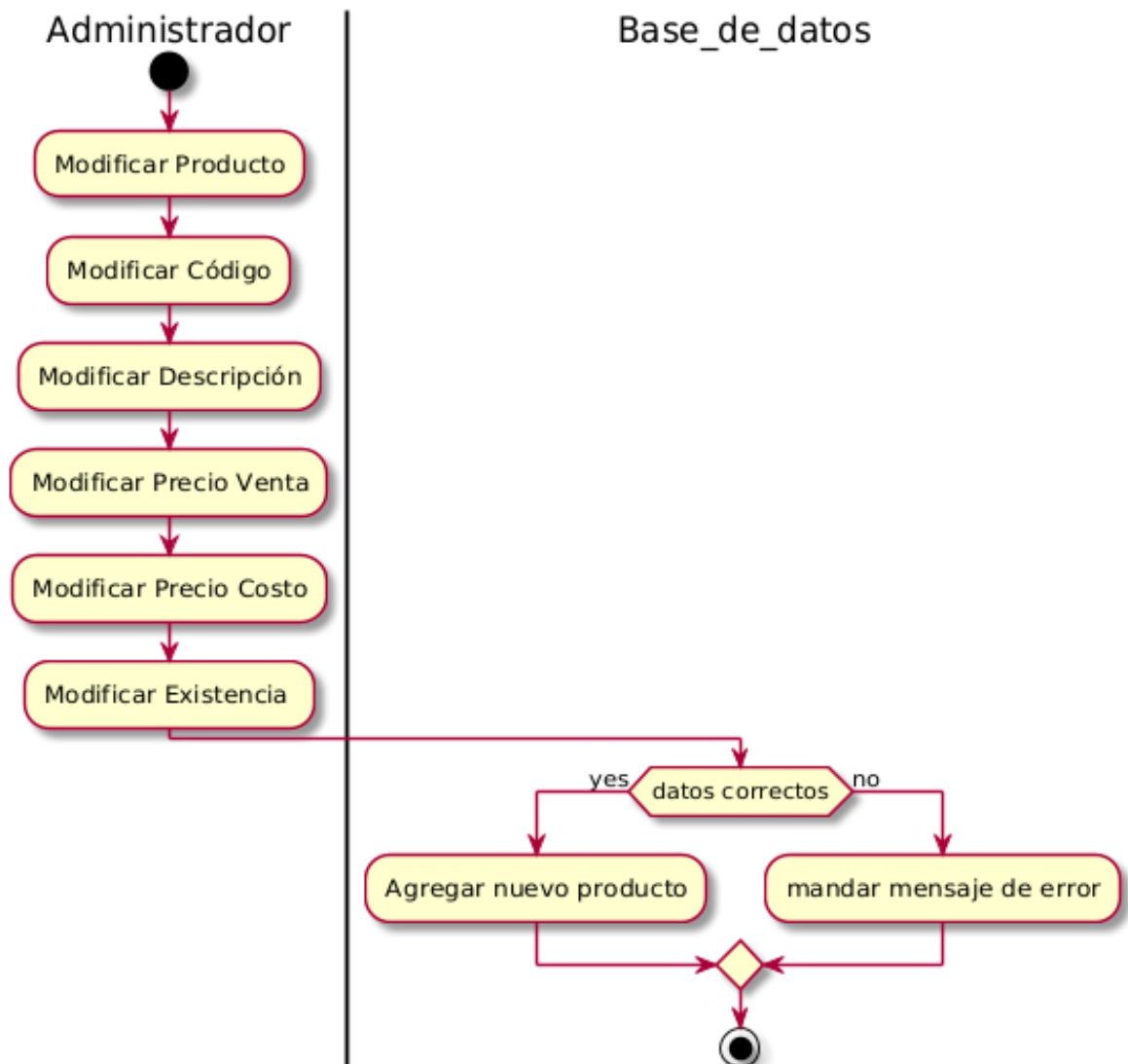


Figura 18.2: Nuevo

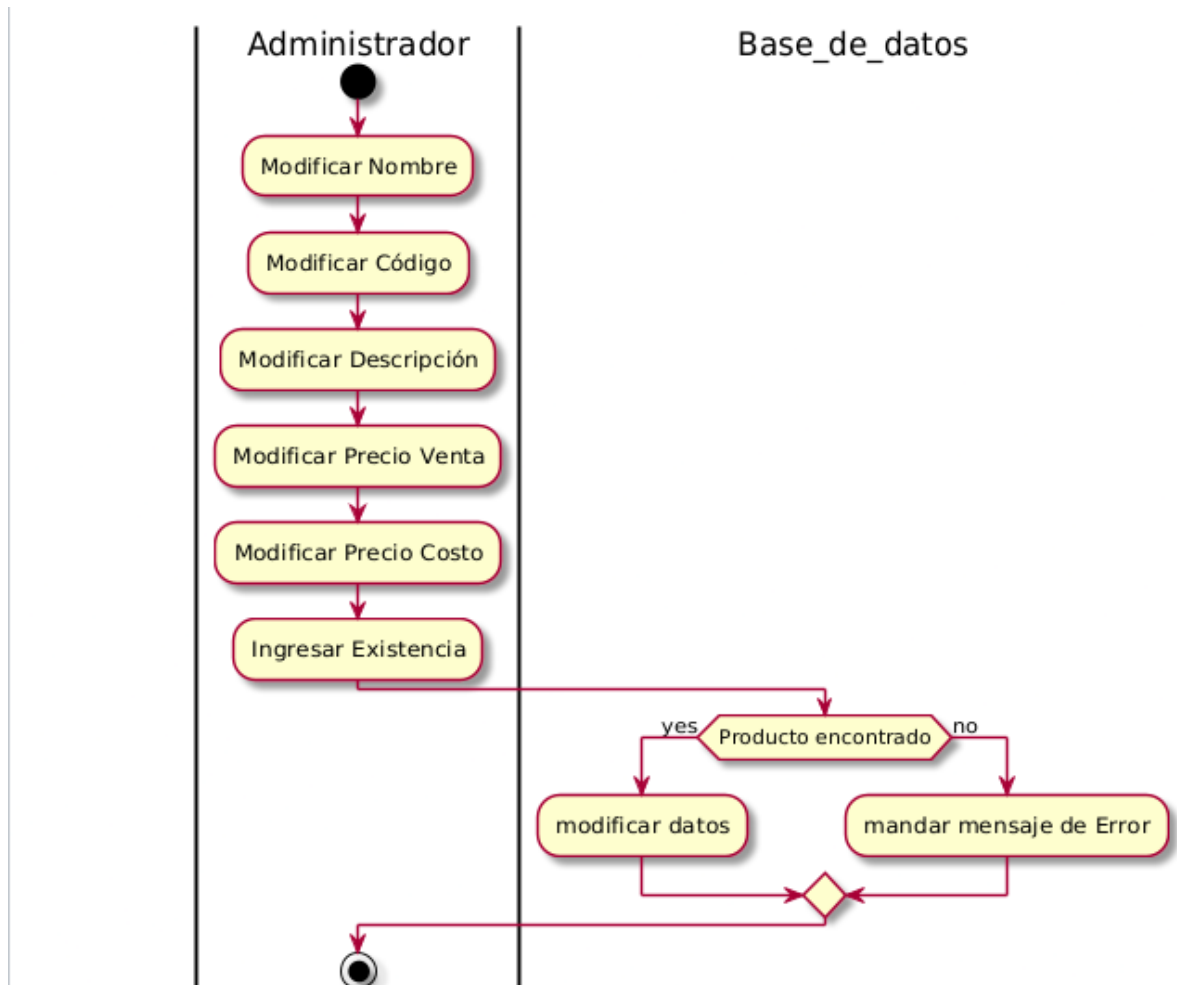


Figura 18.3: Modificar

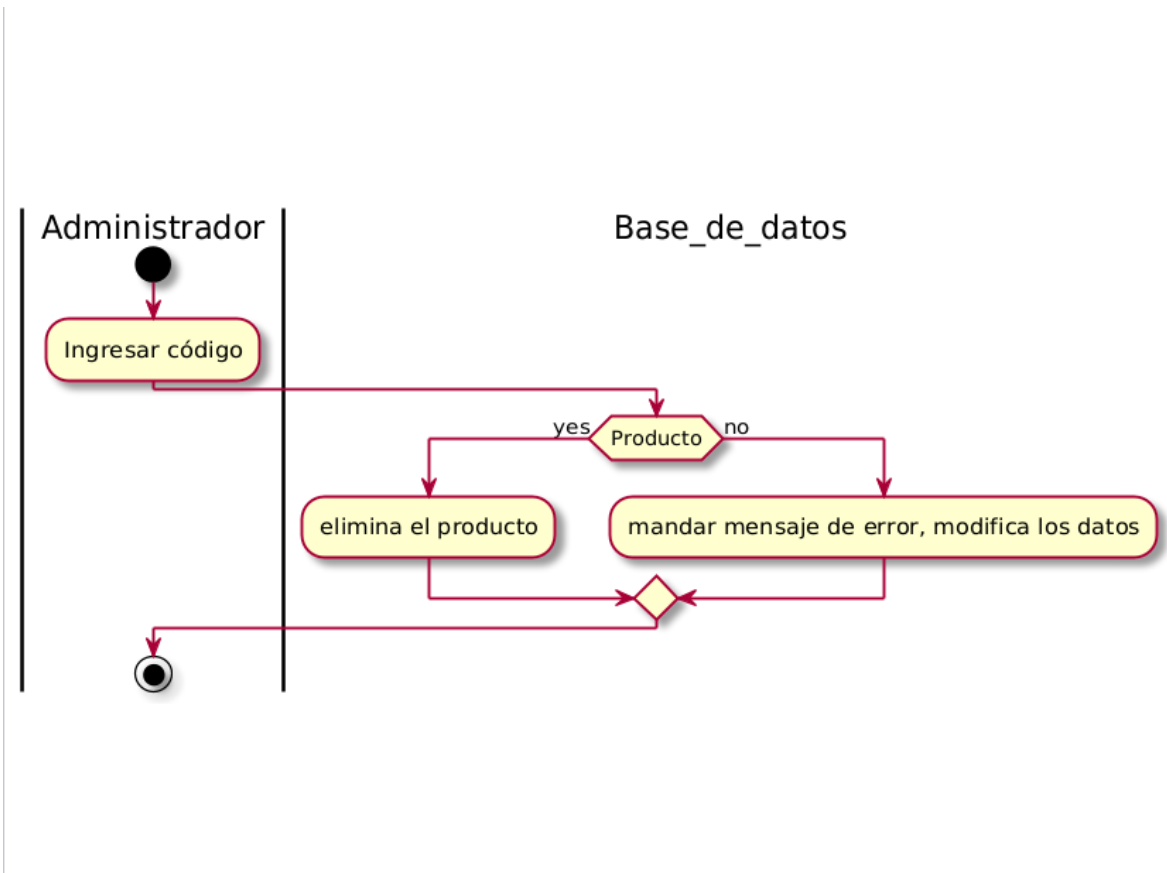


Figura 18.4: Eliminar

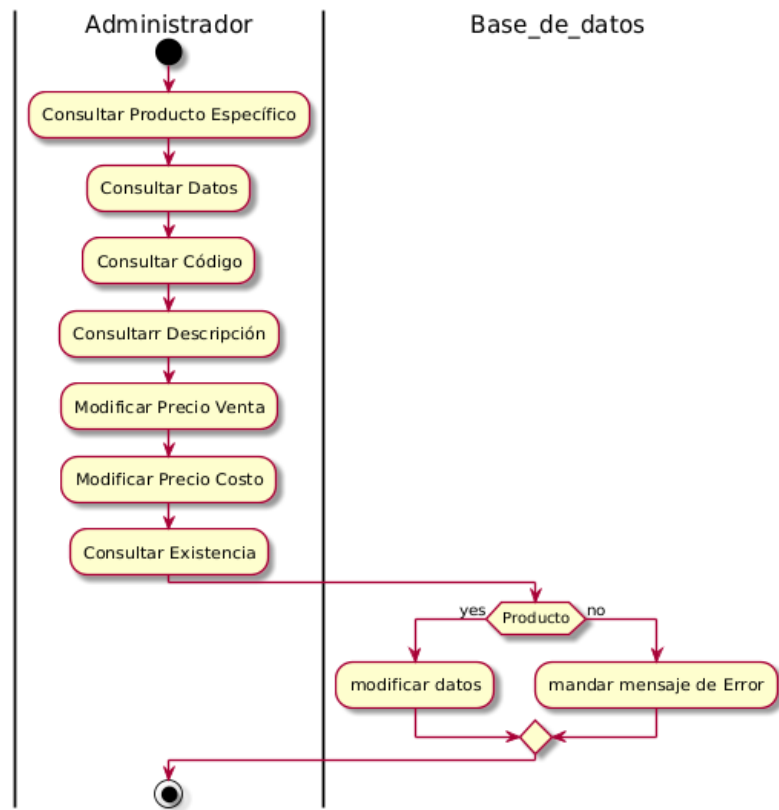


Figura 18.5: Consultar

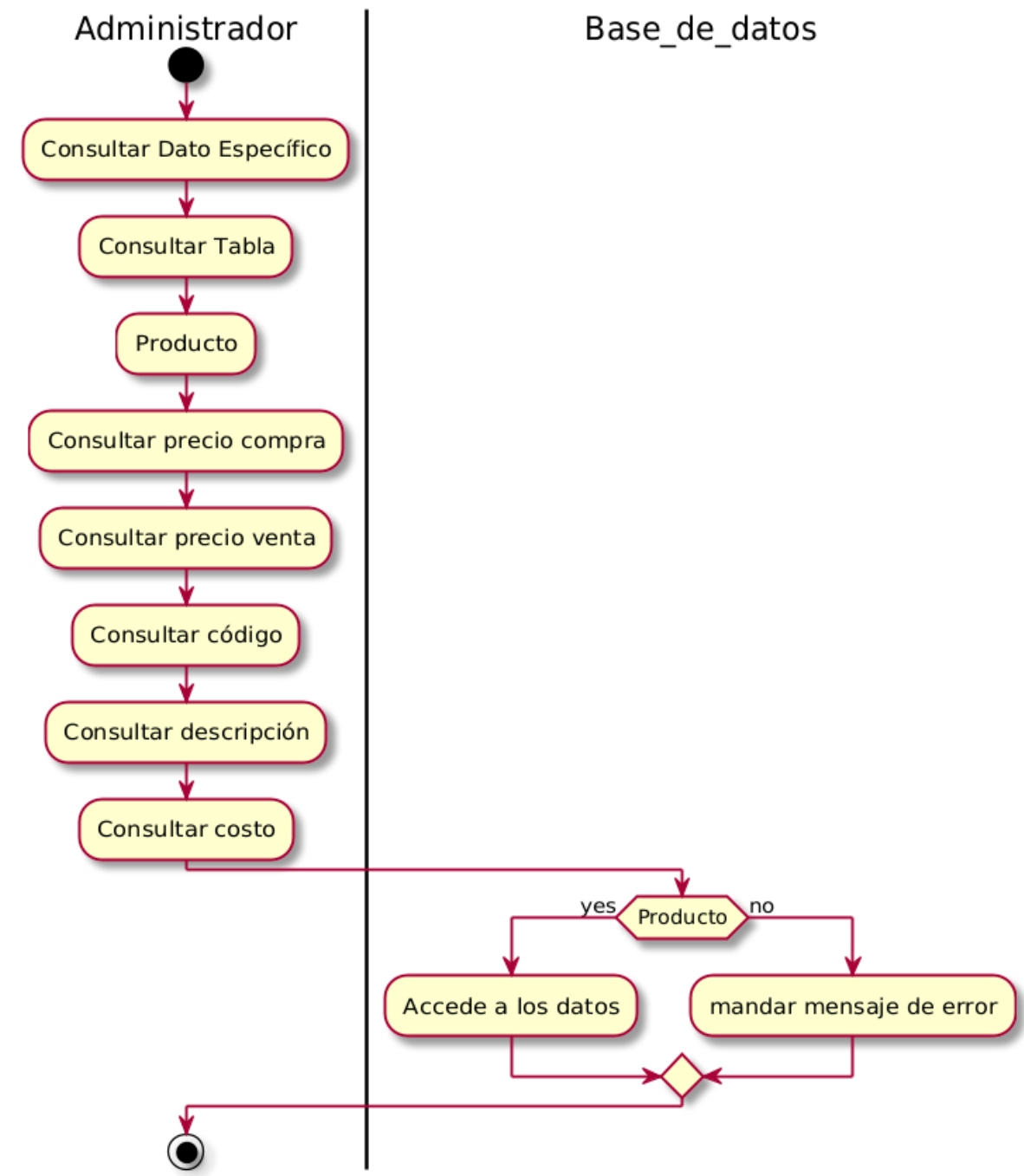


Figura 18.6: Inventario

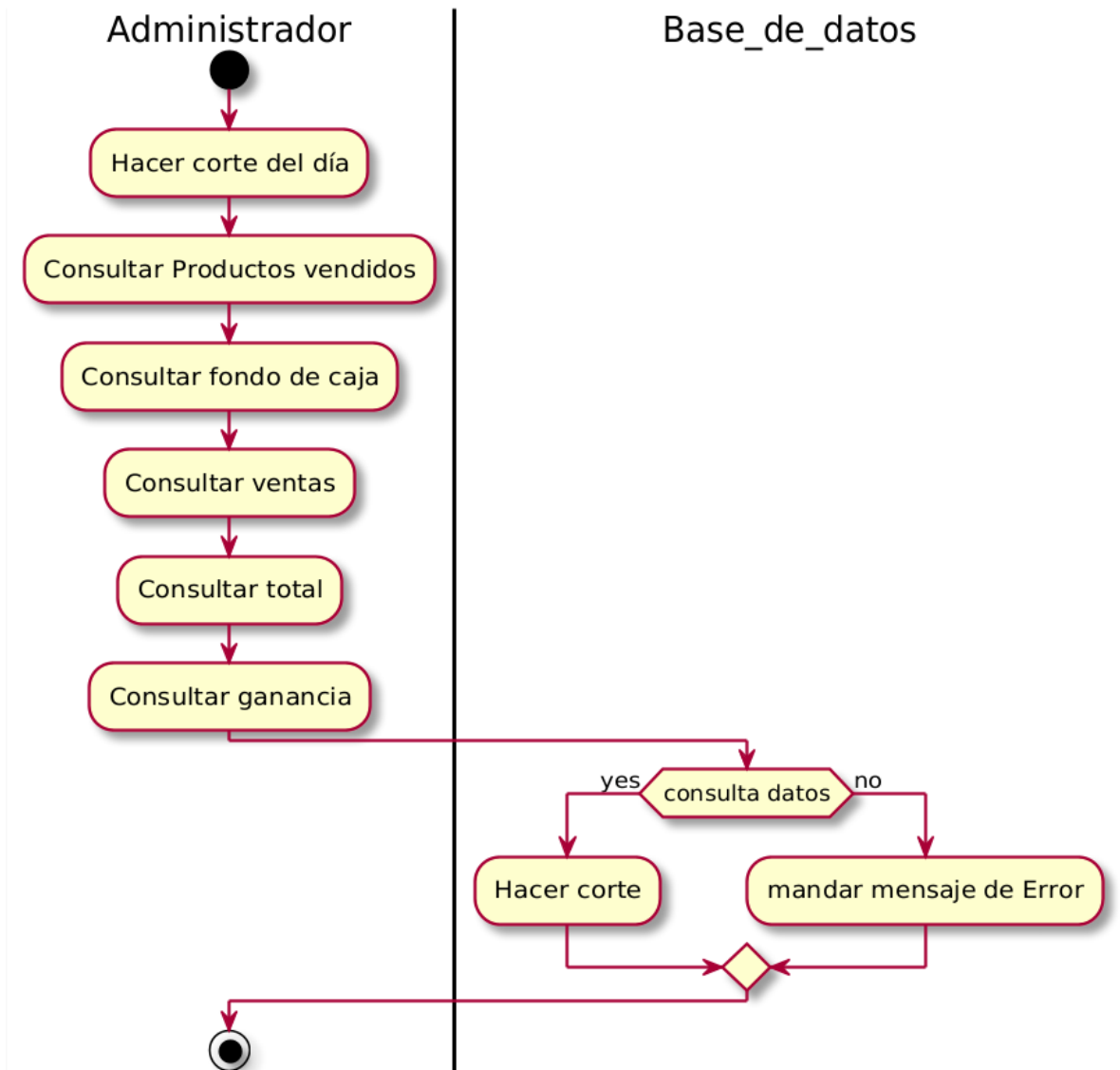


Figura 18.7: Corte

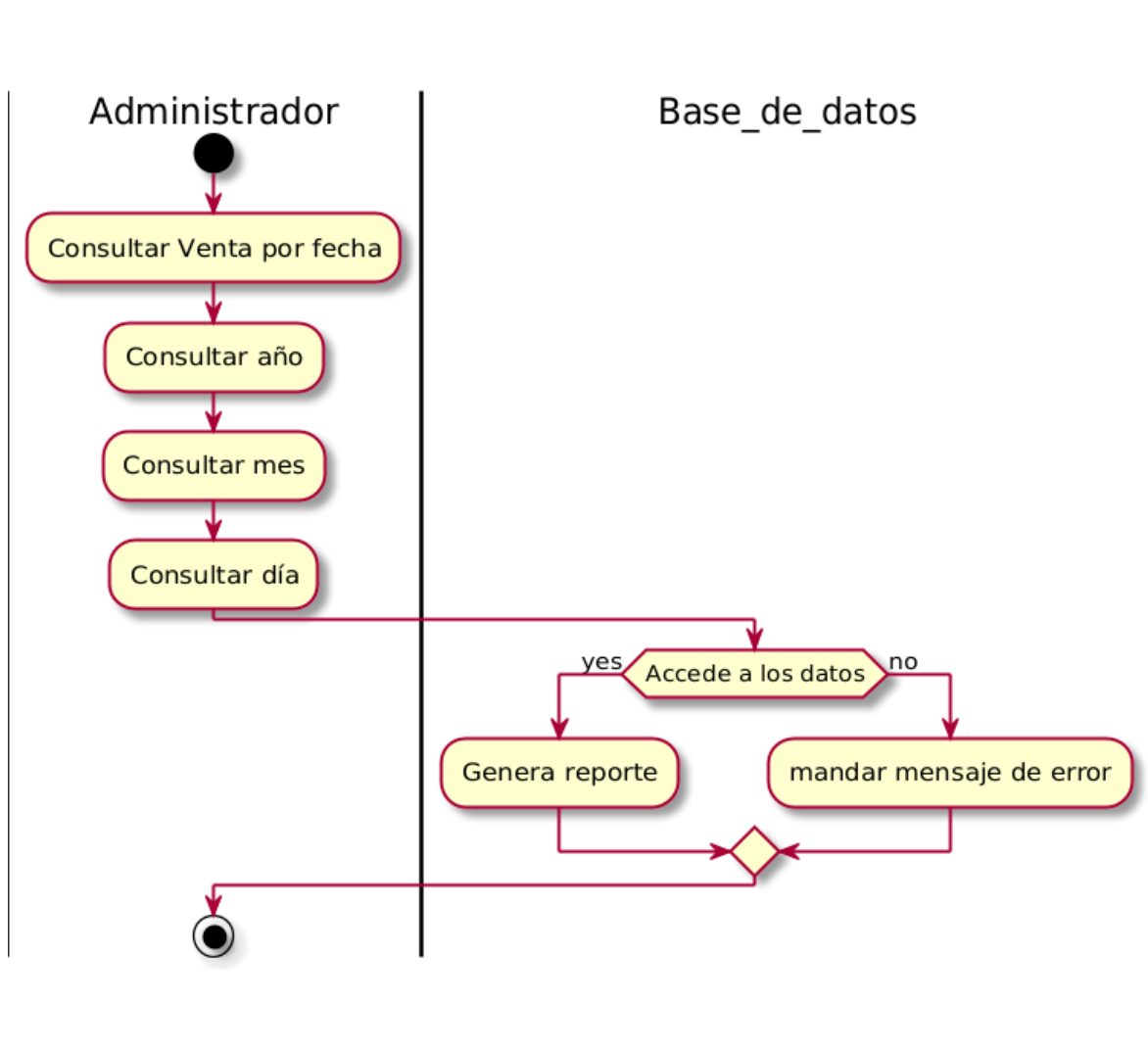


Figura 18.8: Reporte

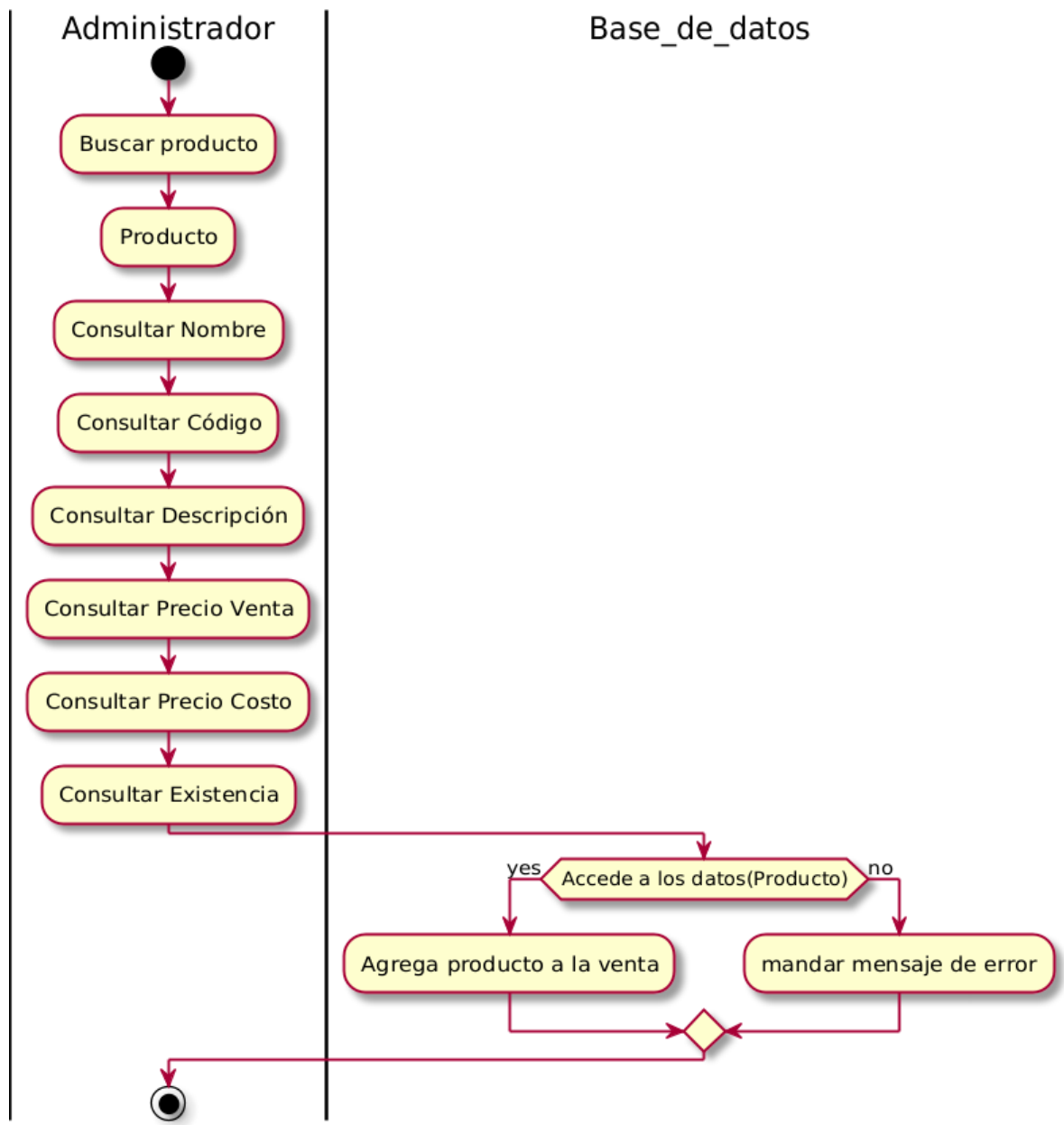


Figura 18.9: Ventas

Capítulo 19

Diagramas de Flujo de Datos

Nivel 0

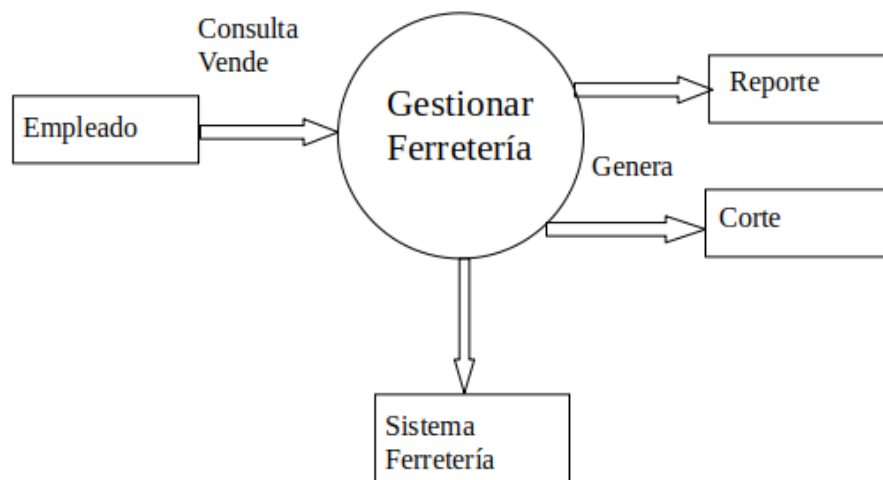


Figura 19.1: Nivel 0

Nivel 0

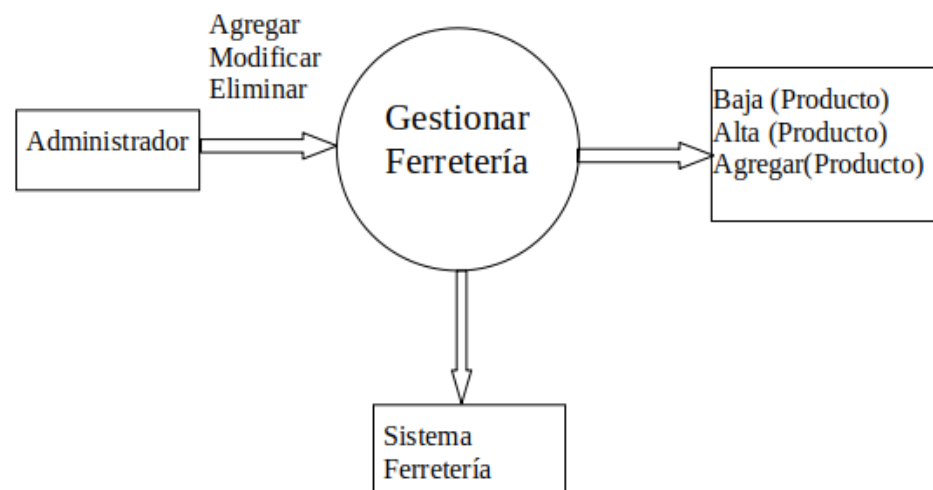


Figura 19.2: Nivel 0.1

Nivel 1

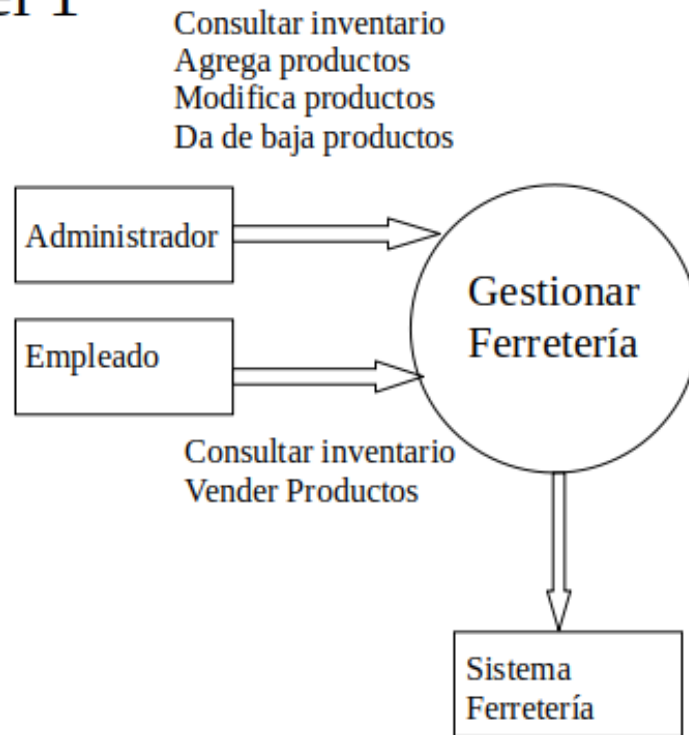


Figura 19.3: Nivel 1

Capítulo 20

Diagrama de Despliegue

Ferretería

```
@startuml
node Inicio as fi
node Login as jl
node Iniciar_sesion as ji
node Administrador as AD
node Empleado as EM
node Ventas as V
node Inventario as I
node Agregar_Nuevo_Producto as NP
node Eliminar_Producto as EP
node Modificar_Producto as MP
node Modificar_Usuarios as US
node Agregar_Usuario as AU
node Eliminar_Usuario as EU
node Corte as CO
node Reporte as RE
node Consultar as CON
```

```
fi--jl
fi--ji
jl--ji
jl--AD
ji--EM
EM--I
EM--V
AD--I
AD--NP
AD--EP
AD--MP
AD--US
AD--AU
AD--EU
```


AD--CO
AD--RE
AD--CON
@endum1

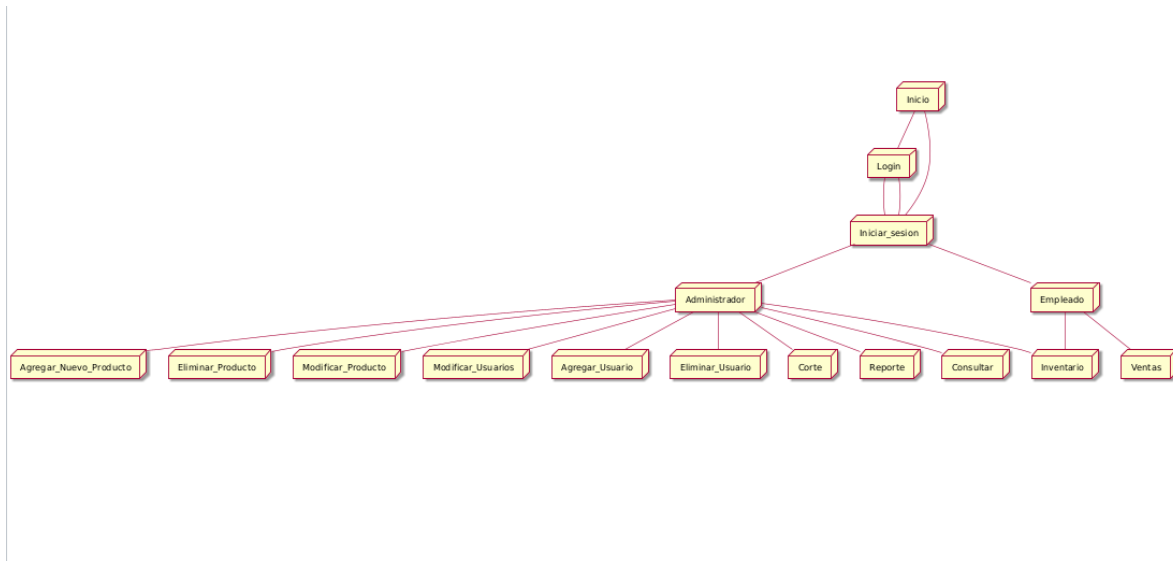


Figura 20.1: Ferretería

Capítulo 21

Diagrama de Componentes

Ferretería

```
@startuml
[BD<<component>>] as b
[Inicio<<component>>] as in

package "Roles"{
[Iniciarsesion<<component>>] as is
[Empleado<<component>>] as em
[Administrador<<component>>] as ad
}

package "Registro"{
[Login<<component>>] as lo
[Registrarse<<component>>] as r
}

package "Productos"{
[AgregarNuevoProducto<<component>>] as anp
[ModificarProducto<<component>>] as mp
[Eliminarproducto<<component>>] as ep
}

package "Usuarios_admin"{
[Agregarusuario<<component>>] as au
[Modificarusuarios<<component>>] as mu
[eliminarrusuario<<component>>] as eu
}

[consultar<<component>>] as cons
[ventas<<component>>] as v
[Inventario<<component>>] as inv
[Corte<<component>>] as cor
in--lo
```

```

lo--ad
lo--em
lo--is
ad--inv
ad--anp
ad--mp
ad--mu
ad--ep
ad--cor
ad--r
ad--au
ad--eu
em--inv
em--v
em--cons

```

```

b--0)-inv
b--0)-ventas
b--0)-usuarios
b--0)-cons

```

```
@enduml
```

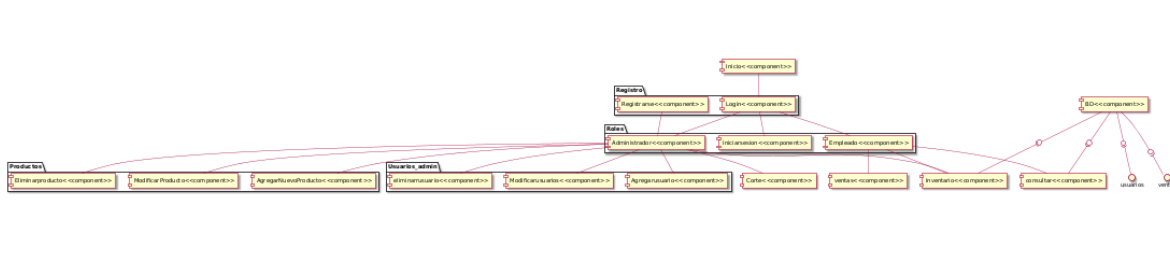


Figura 21.1: Ferretería

Capítulo 22

Diagrama de Objetos

Ferretería

```
@startuml
object Usuarios
Usuarios: roles
object Administrador

Administrador: NombreUsuario:
Administrador: contraseña
object Empleado
Empleado: Usuario
Empleado: Contraseña

object Inventario
Inventario: Productos

object venta
venta: Productos
venta: Precio_costo
venta: Precio_venta
venta: Existencia

object Productos
Productos: Código
Productos: Cantidad
Productos: NuevoProducto
Productos: ModificarProducto
Productos: EliminarProducto

object Corte
Corte: ventas_por_día

object Reporte
Reporte: Ventas_por_mes
```

Reporte: Día
Reporte: Mes
Reporte: Año

object BD
BD--|>venta
BD--|>Corte
BD--|>Reporte
BD--|>Productos
BD--|>Inventario
BD--|>Usuarios

venta--|>Productos
Usuarios--|>Administrador
Usuarios--|>Empleado
Administrador--|>Inventario
Administrador--|>Reporte
Administrador--|>Corte
Empleado--|>venta

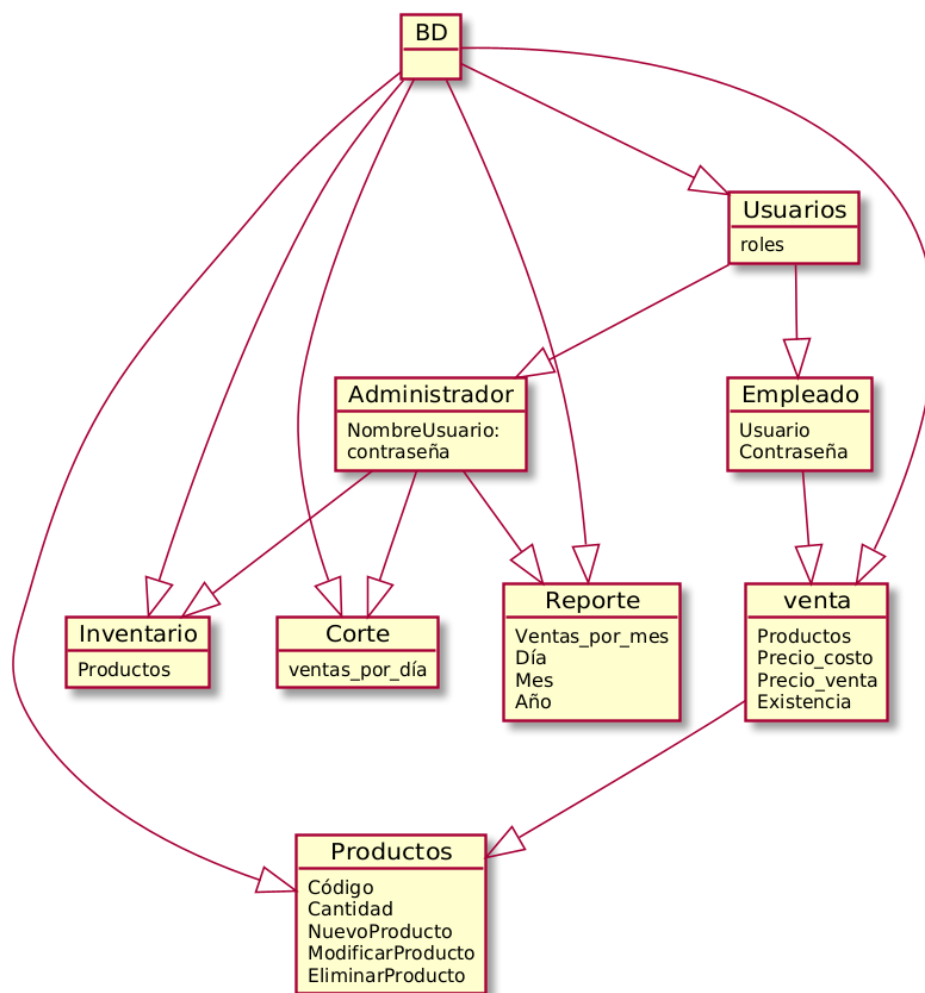


Figura 22.1: Ferretería